República Federativa do Brasil

Ministério do Desenvolvimento, Indústria, Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) PI 0720806-5 B1

(22) Data do Depósito: 18/04/2007

(45) Data de Concessão: 28/03/2023

(54) Título: CORRENTE DE DADOS DE VÍDEO DE QUALIDADE ESCALONÁVEL

(51) Int.Cl.: H04N 19/34; H04N 19/61.

(52) CPC: H04N 19/34; H04N 19/61.

(30) Prioridade Unionista: 18/01/2007 US 60/885,534.

(73) Titular(es): FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V..

(72) Inventor(es): THOMAS WIEGAND; HEIKO SCHWARZ; HEINER KIRCHHOFFER.

(86) Pedido PCT: PCT EP2007003411 de 18/04/2007

(87) Publicação PCT: WO 2008/086828 de 24/07/2008

(85) Data do Início da Fase Nacional: 20/07/2009

(57) Resumo: CORRENTE DE DADOS DE VÍDEO DE QUALIDADE ESCALONÁVEL. Aparelho para gerar uma corrente de dados de vídeo de qualidade escalonável (36) é descrito, o qual compreende meios (42) para codificar um sinal de vídeo (18) usando transformação no âmbito de bloco para obter blocos de transformada (146, 148) de valores de coeficiente de transformação para uma imagem (140) do sinal de vídeo, uma ordem de exploração predeterminada (154, 156, 164, 166) com posições de exploração possíveis sendo definidas entre os valores de coeficiente de transformação dentro dos blocos de transformada, de modo que em cada bloco de transformada, para cada posição de exploração possível, pelo menos um dos valores de coeficiente de transformação dentro do respectivo bloco de transformada pertença à respectiva posição de exploração possível; e meios (44) para formar, para cada uma dentre uma pluralidade de camadas, uma corrente de subdados de vídeo (30; 28, 30) contendo informação de faixa de exploração indicando um subconjunto das possíveis posições de exploração, e informações de coeficiente de transformada nos valores de coeficiente de transformação pertencendo ao subconjunto de possíveis posições de exploração, de modo que o subconjunto de cada camada de qualidade compreenda pelo menos uma possível posição de exploração não compreendida pelo subconjunto de qualquer outra dentre a pluralidade de camadas de qualidade.(...).

"CORRENTE DE DADOS DE VÍDEO DE QUALIDADE ESCALONÁVEL"

Descrição

Histórico

A presente invenção se refere a correntes de dados de vídeo de qualidade escalonável, sua geração e decodificação, tal como a geração e decodificação de correntes de dados de vídeos obtidas pelo uso de transformação com relação a blocos.

O "Joint Video Team, JVT" atual do "ITU-T Video Coding Experts Group" no "ISO/IEC Moving Pictures Expert Group (MPEG)" está especificando atualmente uma extensão escalonável do padrão de codificação de vídeo H.264/MPEG4-AVC. A característica chave da codificação de vídeo escalonável (SVC) em comparação com codificação de camada única convencional é que várias representações de uma fonte de vídeo com diferentes resoluções, taxas de quadro e/ou taxas de bits são providas dentro de uma única corrente de bits. Uma representação de vídeo com uma resolução espaço-temporal e taxa de bits específicas pode ser extraída de uma corrente de bits SVC global por manipulações simples de corrente como descarte de pacotes. Como uma característica importante do projeto SVC, a maioria dos componentes de H.264/MPG4-AVC são usados conforme especificado no padrão. Isto inclui o movimento compensado e a intra-predição, a transformada e codificação de entropia, o desbloqueio assim como o formação de pacotes da unidade NAL (NAL = Camada de Abstração de Rede). A camada base de uma corrente de bits SVC é geralmente codificada em concordância com o H.264-MPEG4-AVC, e dessa maneira

cada padrão em conformidade com o decodificador H.264-MPEG4-AVC é capaz de decodificar a representação da camada base quando ele é provido com uma corrente de dados SVC. Novas ferramentas são adicionadas apenas para suportar escalonabilidade espacial e SNR.

Para escalonabilidade SNR, escalonabilidade de grãos grossos/grãos médios (CGS/MGS) e escalonabilidade de grão fino (FGS) são distinguidas no Rascunho de Trabalho atual. Codificação escalonável SNR de grão grosso ou grão médio é obtida pelo uso de conceitos similares como para escalonabilidade espacial. As imagens de diferentes camadas SNR são codificadas independentemente com parâmetros de movimento específico de camada. Entretanto, de modo a melhorar a eficiência de codificação das camadas intensificadas em comparação com a transmissão simultânea ("simulcast"), mecanismos de predição inter-camadas adicionais foram introduzidos. Estes mecanismos de predição foram feitos intercambiáveis de modo que um codificador pode escolher livremente qual informação de camada base deve ser explorada para uma codificação de camada de intensificação eficiente. Visto que conceitos de predição inter-camadas incorporados incluem técnicas para parâmetro de movimento e predição residual, as estruturas de predição temporal das camadas SNR devem ser temporariamente alinhadas para um uso eficiente da predição inter-camadas. Deve ser observado que todas as unidades NAL para um instante no tempo formam uma unidade de excesso e, dessa maneira, têm que acompanhar umas as outras dentro de uma corrente de bits SVC. As três técnicas de predição inter-camadas a seguir são incluídas no projeto SVC.

A primeira é denominada de predição de movimento

inter-camadas. De modo a empregar dados de movimento de camada base para a codificação de camada de intensificação, um modo de macrobloco adicional foi introduzido na camadas de intensificação SNR. A divisão de macrobloco é obtida pela cópia da divisão do macrobloco co-posicionado na camada base. Os índices de imagem de referência, assim como os vetores de movimento associados são copiados dos blocos de camada base co-posicionados. Adicionalmente, um vetor de movimento da camada base pode ser usado como um preditor de vetor de movimento para os modos de macrobloco convencionais.

A segunda técnica de redução de redundância entre as várias camadas de qualidade é denominada de predição residual inter-camadas. O uso de predição residual inter-camadas é sinalizado por um marcador (residual_prediction_flag) que é transmitido para todos os macroblocos inter-codificados. Quando este marcador for verdadeiro, o sinal da camada base do bloco co-posicionado é usado como predição para o sinal residual do macrobloco atual, de modo que apenas o sinal de diferença correspondente é codificado.

Finalmente, a intra-predição inter-camadas é usada de modo a explorar redundância entre as camadas. Neste modo intramacrobloco, o sinal de predição é construído pelo sinal de reconstrução co-posicionado da camada base. Para a intra-predição inter-camadas é geralmente requerido que camadas bases sejam completamente decodificadas, incluindo as operações complexas de computação de predição de compensação de movimento e eliminação de bloco. Entretanto, foi mostrado que este problema pode ser contornado quando a intra-predição inter-camadas é restrita

àquelas partes da imagem da camada inferior que são intra-codificadas. Com esta restrição, cada camada alvo suportada pode ser decodificada com uma alça de compensação de movimento única. Este modo de decodificação de alça única é obrigatório na extensão escalonável H.264-MPEG4-AVC.

Visto que a intra-predição inter-camadas pode apenas ser aplicada quando o macrobloco co-posicionado é intracodificado e a predição de movimento inter-camadas com inferência do tipo de macrobloco pode ser aplicada apenas quando o macrobloco de camada base é intercodificado, ambos os modos são sinalizados por meio de um elemento de sintaxe único base_mode_flag em um nível de macrobloco. Quando este marcador for igual a 1, intra-predição inter-camadas é escolhida quando o macrobloco de camada base for intracodificado. Do contrário, o modo de macrobloco, assim como os índices de referência e vetores de movimento, são copiados do macrobloco de camada base.

De modo a suportar uma granularidade mais fina do que codificação CGS/MGS, pedaços de refinamento progressivo assim denominados foram introduzidos, os quais permitem codificação escalonável SNR granular mais fina (FGS). Cada pedaço de refinamento progressivo representa um refinamento do sinal residual que corresponde a uma bissecção do tamanho das etapas de quantização (aumento QP de 6). Estes sinais são representados de uma maneira que apenas uma transformada inversa única tem que ser executada para cada bloco de transformada no lado do decodificador. A ordem de níveis de coeficiente de transformada nos pedaços de refinamentos progressivos permite que as unidades NAL correspondentes sejam truncadas em qualquer ponto de

alinhamento de byte arbitrário, de modo que a qualidade da camada base SNR pode ser refinada de uma maneira granular fina. Adicionalmente a um refinamento do sinal residual, é também possível transmitir um refinamento de parâmetros de movimento como parte dos pedaços de refinamento progressivo.

Uma desvantagem da codificação FGS no desenho SVC atual é que ela aumenta significativamente a complexidade do decodificador em comparação com a codificação CGS/MGS. Por um lado, os coeficientes de transformada em um pedaço de refinamento progressivo são codificados usando várias explorações nos blocos de transformada, e em cada exploração apenas uns poucos níveis de coeficiente de transformada são transmitidos. Para o decodificador isto aumenta a complexidade visto que uma largura de banda de memória mais elevada é requerida, porque todos os níveis de coeficiente de transformada de diferentes explorações necessitam ser coletados antes que a transformada inversa possa ser executada. Por outro lado, o processo de análise para pedaços de refinamento progressivo é dependente dos elementos de sintaxe dos pedaços de camada base correspondentes. A ordem dos elementos de sintaxe, assim como as tabelas de palavras de código para codificação VLC ou a seleção do modelo de probabilidade para codificação aritmética, dependem dos elementos de sintaxe na camada base. Isto aumenta adicionalmente a largura de banda de memória requerida para decodificação, visto que os elementos de sintaxe da camada base necessitam ser acessados durante a análise da camada de intensificação.

Adicionalmente, a propriedade especial de pedaços de refinamento progressivo de poderem ser truncados é difícil de

usar nas redes de comutação de pacote atuais. Usualmente, um dispositivo de rede ciente da mídia irá tanto enviar ou descartar um pacote de uma corrente de bits escalonável. E o único erro que será visível na camada de aplicação é uma perda de pacote.

Portanto, não apenas em vista do H.264-MPEG4-AVC acima, mas também com outras técnicas de compressão de vídeo, seria desejável ter um esquema de codificação que fosse mais bem adaptado às necessidades de hoje, mostrando perda de pacote ao invés de problemas de truncamento no âmbito de bytes.

Sumário

De acordo com uma configuração da presente invenção, um aparelho para gerar uma corrente de dados de vídeo de qualidade escalonável compreende meios para codificação de um sinal de vídeo usando transformação no âmbito de bloco para obter blocos de transformada de valores de coeficiente de transformação para uma imagem do sinal de vídeo, uma ordem de exploração predeterminada com possíveis posições de exploração sendo definidas entre os valores de coeficiente de transformação dentro dos blocos de transformada, de modo que em cada bloco de transformada, para cada posição de exploração possível, pelo menos um dos valores de coeficiente de transformação dentro do bloco de transformada respectivo, pertence à respectiva posição de exploração possível; e meios para formar, para cada uma dentre uma pluralidade de camadas de qualidade, uma corrente de subdados de vídeo contendo informação de faixa de exploração indicando um subconjunto das possíveis posições de exploração, e informação de coeficiente de transformada nos valores de coeficiente de transformação que pertencem ao subconjunto de possíveis posições

de exploração, de modo que o subconjunto de cada camada de qualidade compreende pelo menos uma posição de exploração possível não compreendida pelo subconjunto de qualquer outra dentre a pluralidade de camadas de qualidade.

Adicionalmente, de acordo com uma configuração da presente invenção, um aparelho para reconstruir um sinal de vídeo de uma corrente de dados de video de qualidade escalonável compreendendo, para cada uma dentre uma pluralidade de camadas de qualidade, uma corrente de subdados de vídeo, compreende meios para análise das correntes de subdados de vídeo da pluralidade de camadas de qualidade, para obter, para cada camada de qualidade, uma informação de faixa de exploração e informação de coeficiente de transformada nos valores de coeficiente de transformação de diferentes blocos de transformada, uma ordem de exploração predeterminada com possíveis posições de exploração sendo definida entre os valores de coeficientes de transformação dentro dos blocos de transformada, de modo que em cada bloco de transformada, para cada possível posição de exploração, pelo menos um dos valores de coeficiente de transformação dentro do respectivo bloco de transformada pertence à respectiva posição de exploração possível, e a informação de faixa de exploração indicando um subconjunto das possíveis posições de exploração, meios para, usando a informação de faixa de exploração, para cada camada de qualidade, construir os blocos de transformada pela associação dos valores de coeficiente de transformação dos respectivos blocos de transformada a partir da informação de coeficiente de transformada para o subconjunto das possíveis posições de exploração; e meios para reconstruir uma imagem do sinal de vídeo por uma retro-

transformação dos blocos de transformada.

Breve Descrição dos Desenhos

Com base nas Figuras, algumas configurações da presente invenção são descritas em maiores detalhes abaixo. Em especial,

A Figura 1 mostra um diagrama de bloco de um codificador gerando uma corrente de dados de vídeo de qualidade escalonável de acordo com uma configuração;

A Figura 2 mostra um diagrama de bloco de um codificador híbrido de camada mais elevada da Figura 1 de acordo com uma configuração;

A Figura 3 mostra uma diagrama de bloco de um codificador híbrido de camada base da Figura 1 de acordo com uma configuração;

A Figura 4 mostra uma diagrama de bloco de uma unidade de codificação de camada da camada de qualidade superior da Figura 1 de acordo com uma configuração;

A Figura 5 mostra uma diagrama esquematizado ilustrando a estrutura de uma imagem, assim como sua transformação no âmbito de bloco de acordo com uma configuração;

As Figuras 6a-6g mostram diagramas esquematizados de uma porção explorada de um bloco de transformada e sua divisão em subcamadas de acordo com várias configurações;

A Figura 7 mostra um diagrama esquematizado ilustrando a construção de correntes de subdados de acordo com uma configuração;

A Figura 8 mostra um pseudocódigo ilustrando a codificação dos níveis de coeficiente de transformada que

pertencem a uma corrente de subdados específica de acordo com uma configuração;

A Figura 9 mostra um pseudocódigo ilustrando um outro exemplo para codificação dos níveis de coeficiente de transformada que pertencem a uma corrente de subdados específica;

A Figura 10 mostra um diagrama de bloco de um decodificador de acordo com uma outra configuração; e

A Figura 11 mostra um diagrama de bloco de uma configuração para a unidade de decodificação da Figura 10.

Descrição Detalhada dos Desenhos

A Figura 1 mostra um codificador para gerar uma corrente de bits de qualidade escalonável. Exemplificativamente, o codificador 10 da Figura 1 é dedicado para a geração de uma corrente de dados escalonável que suporta duas camadas espaciais diferentes e camadas N+1 SNR. Para este fim, o codificador 10 é estruturado em uma parte de camada base 12 e uma parte de camada de intensificação espacial 14. Uma unidade de redução de qualidade 16 do codificador 10 recebe o vídeo original ou de maior qualidade 18 representando uma seqüência de imagens 20 e reduz sua qualidade - no sentido de resolução espacial no exemplo da Figura 1 - para obter uma versão de qualidade inferior 22 do vídeo original 18 consistindo de uma seqüência de imagens 24, a versão de qualidade inferior 22 sendo inserida na parte da camada base 12.

A unidade de redução da qualidade 16 executa, por exemplo, uma subamostragem das imagens por um fator de subamostragem de 2, por exemplo, Entretanto, deve ser entendido que embora a figura 1 mostre um exemplo que suporta duas camadas espaciais 12, 14, a configuração da Figura 1 pode prontamente ser

aplicada a aplicações onde a redução da qualidade executada entre o vídeo original 18 e o vídeo de qualidade inferior 22 não compreende uma subamostragem, mas, por exemplo, uma redução na profundidade de bit da representação dos valores de pixel, ou a unidade de redução da qualidade simplesmente copia o sinal de entrada para o sinal de saída.

Enquanto a parte de camada base 12 recebe o vídeo de qualidade inferior 22, o vídeo original 18 é inserido na parte de qualidade superior 14, com ambas as partes 12, 14 executando uma codificação híbrida no vídeo respectivamente inserido. A parte de camada base 12 recebe o vídeo de qualidade inferior 22 e gera uma corrente de bits de camada base 26. Por outro lado, a parte de camada de qualidade superior 14 recebe em sua entrada o vídeo original 18 e gera, além de uma corrente de bits de camada de intensificação 28, correntes de bits de camada de refinamento N SNR 30. A geração e o inter-relacionamento entre correntes de bits 28 e 26 serão descritos em maiores detalhes abaixo. Apenas como uma medida de precaução, é observado que a parte da camada base 12 poderia também acompanhar a corrente de bits de camada base 26 por várias camadas de refinamento SNR 32. Entretanto, de modo a facilitar a ilustração dos princípios da presente configuração, é pressuposto que a escalonabilidade SNR é restrita à parte de camada de intensificação 14. Entretanto, a discussão a seguir revelará que a funcionalidade descrita abaixo com relação à parte de camada de qualidade superior 14 com respeito às camadas de refinamento SNR é prontamente transferível para a parte da camada base 12. Isto é indicado na Figura 1 por uma linha pontilhada 32.

Todas as correntes de bits de 26 a 32 são

inseridas em um multiplexador 34 que gera uma corrente de bits escalonável 36 a partir das correntes de dados em sua entrada, eventualmente arranjada em pacotes, como será descrito em maiores detalhes abaixo.

Internamente, a parte da camada base 12 compreende um codificador híbrido 38 e uma unidade de codificação de camada 40 conectados em série, na ordem mencionada, entre a entrada à qual o vídeo de qualidade inferior 24 é aplicado, por um lado e o multiplexador 34, por outro lado. Similarmente, a parte de camada de qualidade superior 14 compreende um codificador híbrido 42 e uma unidade de codificação de camada 33 conectados entre a entrada à qual o vídeo original 18 é aplicado e o multiplexador 44. Cada codificador híbrido 42 e 38, respectivamente, codifica seu sinal de entrada de vídeo por codificação híbrida, isto é, predição compensada por movimento é usada juntamente com transformação no âmbito de bloco do residual de predição. Dessa maneira, cada codificador híbrido 38 e 42, respectivamente, envia dados de informação de movimento 46 e 48, respectivamente, assim como dados residuais 50 e 52, respectivamente, para a entrada da unidade de codificação da camada subseqüente 40 e 44, respectivamente.

Naturalmente, existe redundância entre os dados de movimento 46 por um lado e 48 por outro lado, assim como para os dados residuais 50 e 52. Esta redundância inter-camadas é explorada pelo codificador híbrido 42. Em especial, em uma base de macroblocos, o codificador híbrido 42 pode escolher entre várias opções de predição inter-camadas. Por exemplo, o codificador híbrido 42 pode decidir usar ou adotar os dados de movimento da

camada base 46 como os dados de movimento 48 para a camada de qualidade superior. Alternativamente, o codificador híbrido 42 pode decidir usar os dados de movimento de camada base 46 como preditor para os dados de movimento 48. Como uma alternativa adicional, o codificador híbrido 42 pode codificar os dados de movimento 48 completamente de outra forma, isto é, independente dos dados de movimento da camada base.

Similarmente, o codificador híbrido 42 pode codificar os dados residuais 42 para a camada de qualidade superior de forma preditiva como o residual de predição em relação aos dados residuais de camada base 50 como um preditor.

Entretanto, o codificador híbrido 42 pode também usar uma reconstrução do teor da imagem da camada base como um preditor para o conteúdo da imagem dos dados do vídeo original, de modo que neste caso dados de movimento e/ou dados residuais 48 e 52, respectivamente, codificam meramente o residual em relação aos dados de camada base reconstruída. Como será descrito com relação à Figura 2, a informação de imagem de camada base reconstruída pode ser recebida pelo codificador híbrido de camada base 38 ou uma unidade de reconstrução dedicada 54 acoplada entre a unidade de codificação de camada base 40 e um codificador híbrido de camada de qualidade superior 42.

A seguir, uma estrutura interna e a funcionalidade dos codificadores híbridos 38 e 42, assim como a unidade de codificação de camada 44, serão descritos em maiores detalhes. Com relação à unidade de codificação de camada 40, a seguir é pressuposto que a mesma gera meramente corrente de dados de camada base 26. Entretanto, conforme indicado acima, uma

alternativa de uma configuração de acordo com a qual a unidade de codificação de camada 40 também gera correntes de dados de camada de refinamento SNR 32, é prontamente derivável da descrição a seguir com respeito à unidade de codificação de camada 44.

Primeiramente, a estrutura interna e a funcionalidade do codificador híbrido de camada base 38 são descritas. Conforme mostrado na Figura 3, o codificador híbrido de camada base 38 compreende uma entrada 56 para receber os sinais de vídeo de qualidade inferior 24, uma saída 58 para os dados de movimento 46, uma saída 60 para os dados residuais 50, uma saída 62 para acoplamento dos dados de movimento 58 ao codificador híbrido 42, uma saída 64 para acoplamento de dados de imagem de camada base reconstruídos ao codificador híbrido 42, e uma saída 66 para acoplamento de dados residuais 50 ao codificador híbrido 42.

Internamente, o codificador híbrido 38 compreende uma unidade de transformação 68, uma unidade de retro-transformação 70, um subtrator 72, um adicionador 74, e uma unidade de predição de movimento 76. O subtrator 72 e a unidade de transformação 68 são acoplados, na ordem mencionada, entre a entrada 56 e a saída 60. O subtrator 72 subtrai da entrada o conteúdo de vídeo predito por movimento recebido da unidade de predição de movimento 76 e envia o sinal de diferença para a unidade de transformação 68. A unidade de transformação 68 executa uma transformação em âmbito de bloco no sinal de diferença/residual juntamente com, opcionalmente, uma quantização dos coeficientes de transformada. O resultado da transformação é enviado pela unidade de transformação 68 para a saída 60, assim

como para uma entrada da unidade de retro-transformação 70. A unidade de retro-transformação 70 executa uma transformação inversa nos blocos de transformada dos coeficientes de transformação com, eventualmente, uma desquantização precedente. O resultado é um sinal residual reconstruído que é, pelo adicionador 74, combinado aditivamente com o conteúdo de vídeo predito por movimento enviado pela unidade de predição de movimento 76. O resultado da adição executada pelo adicionador 174 é um vídeo reconstruído na qualidade base. A saída do adicionador 74 é acoplada a uma entrada da unidade de predição de movimento 76, assim como a uma saída 64. A unidade de predição de movimento 76 executa uma predição compensada por movimento baseada nas imagens reconstruídas, de modo a predizer outras imagens da entrada de vídeo para a entrada 56. A unidade de predição de movimento 76 produz, enquanto executa predição de movimento, incluindo dados de movimento, por exemplo, vetores de movimento e índices de referência de imagem de movimento e envia estes dados de movimento de modo para a saída 62, assim como para a saída 58. A saída da unidade de transformação 68 é também acoplada à saída 66 de modo a enviar os dados residuais de transformada para o codificador híbrido 42 da camada de qualidade superior. Como já mencionado acima, as funcionalidades de ambos os codificadores híbridos 38 e 42 da Figura 1 são similares entre si. Entretanto, o codificador híbrido 42 da camada de qualidade superior também usa predição inter-camadas. Dessa maneira, a estrutura do codificador híbrido 42 mostrada na Figura 2 é similar à estrutura do codificador híbrido 38 mostrado na Figura 3. Em especial, o codificador híbrido 42 compreende uma entrada 86 para o sinal de vídeo

original 18, uma saída 88 para os dados de movimento 48, uma saída 90 para os dados residuais 52, e três entradas 92, 94 e 96 para serem acopladas com as respectivas saídas 62, 64 e 66 do codificador híbrido de camada base 38. Internamente, o codificador híbrido 42 compreende dois comutadores ou seletores 98 e 100 para conectar um dos dois caminhos 102 e 104 entre a entrada 86 e a saída 90. Em especial, o caminho 104 compreende um subtrator 106, uma unidade de transformação 108 e um codificador de predição residual 110 sendo conectado, na ordem mencionada, entre a entrada 86 e a saída 90 por meio dos comutadores 98 e 100. O subtrator 106 e a unidade de transformação 108 formam, juntamente com a unidade de retro-transformação 112, um adicionador 114 e uma unidade de predição de movimento 116, uma alça de predição tal como aquela formada pelos elementos 68 a 76 no codificador híbrido 38 da Figura 3. Consequentemente, na saída da unidade de transformação 108, resulta uma versão transformada dos residuais preditos por movimento que é inserida no codificador de predição residual 110. O codificador de predição residual 110 é também conectado à entrada 96 de modo a receber os dados residuais de camada base. Pelo uso destes dados residuais de camada base como um preditor, o codificador de predição residual 110 codifica uma parte da saída de dados residuais pela unidade de transformação 108 como um residual de predição relativo aos dados residuais na entrada 96. Por exemplo, o codificador de predição 110 amostra de forma ascendente os dados residuais de camada base e subtrai os dados residuais amostrados de forma ascendente da saída de dados residuais pela unidade de transformação 108. Obviamente, o codificador de preditor residual 110 pode executar a predição

apenas para uma parte dos dados residuais enviados pela unidade de transformação 108. Outros caminhos passam o codificador de predição 110 inalterados. A granularidade destas partes pode ser de macroblocos. Em outras palavras, a decisão quanto a se os dados residuais na entrada 96 podem ser usados como um preditor ou não, pode ser conduzida em uma base de macrobloco e o resultado da decisão pode ser indicado por um elemento de sintaxe respectivo residual_prediction_flag.

Similarmente, o codificador híbrido 42 compreende um codificador de predição de parâmetro de movimento 118 de modo a receber os dados de movimento na entrada 92, a partir da camada base, assim como a informação de movimento obtida da unidade de predição de movimento 116 e comutadores, em uma base de macrobloco, entre a passagem dos dados de movimento da unidade de predição de movimento 116 inalterados para a saída 88, ou codificando por predição os dados de movimento pelo uso da informação de movimento da camada base na entrada 92 como um preditor. Por exemplo, o codificador de predição de parâmetro de movimento 118 pode codificar vetores de movimento a partir da unidade de predição de movimento 116 como vetores de deslocamento em relação aos vetores de movimento contidos nos dados de movimento da camada base na entrada 92. Alternativamente, codificador de predição de parâmetro de movimento 118 passa a informação da camada base a partir da entrada 92 para a unidade de predição de movimento 116 a ser usada para a predição de movimento na camada de qualidade superior. Neste caso, nenhum dado de movimento tem que ser transmitido para a respectiva porção do sinal de vídeo da camada de qualidade superior. Como uma

alternativa adicional, o codificador de predição de parâmetro de movimento 118 ignora a existência dos dados de movimento na entrada 92 e codifica os dados de movimento a partir da unidade de predição de movimento 116 diretamente para a saída 88. A decisão entre estas possibilidades é codificada na corrente de bits de escalonabilidade de qualidade resultante.

Finalmente, o codificador de predição 120 é provido no caminho 102 e acoplado com a entrada 94. O codificador de predição 120 prediz porções da camada de qualidade superior com base nas porções respectivas do sinal de vídeo da camada base reconstruída, de modo que na saída do codificador de predição 120, meramente o respectivo residual ou diferença é enviado. O codificador de predição 120 também opera em uma base no âmbito de macrobloco em cooperação com os comutadores 98 e 100.

Como pode ser visto da Figura 4, a unidade de codificação de camada 44 da camada de qualidade superior compreende uma entrada 122 para receber os coeficientes de transformada de dados residuais da saída 90 e uma entrada 124 para receber os dados de movimento da saída 88. Uma unidade de distribuição 126 recebe os coeficientes de transformação e distribui os mesmos para várias camadas de intensificação. Os coeficientes de transformação distribuídos dessa maneira são enviados para uma unidade de formação 128. Juntamente com os coeficientes de transformação distribuídos, a unidade de formação 128 recebe os dados de movimento da entrada 124. A unidade de formação 128 combina ambos os dados e formas, com base nestas entradas de dados, a corrente de dados de camada de intensificação de ordem zero 28, assim como correntes de dados de camada de

refinamento 30.

De modo a permitir uma descrição mais detalhada da funcionalidade da unidade de distribuição 126 e da unidade de formação 128, a seguir a base de bloco subjacente à transformação executada pela unidade de transformação 108 e seu inter-relacionamento com a distribuição executada pela unidade de distribuição 126 será descrita em maiores detalhes com relação à Figura 5. A Figura 5 representa uma imagem 140. A imagem 140 é, por exemplo, parte de dados de vídeo de qualidade superior 18 (Figura 1). Dentro da imagem 140, os pixéis são, por exemplo, arranjados em linhas e colunas. A imagem 140 é, por exemplo, dividida em macroblocos 142, que podem também ser arranjados regularmente em linhas e colunas. Cada macrobloco 142 pode, por exemplo, cobrir espacialmente uma área de imagem retangular de modo a compreender, por exemplo, amostras 16x16, por exemplo, do componente luma da imagem. Para ser ainda mais preciso, os macroblocos 142 podem ser organizados em pares de macroblocos. Em especial, o par adjacente verticalmente de macroblocos 142 pode formar um par de macroblocos e pode assumir, espacialmente, uma região de par de macrobloco 144 da imagem 140. Em uma base de par de macroblocos, o codificador híbrido 42 (Figura 1) pode manipular os macroblocos 142 dentro da respectiva região 144 no modo de campo ou modo de moldura. No caso de modo de campo, o vídeo 18 é presumido conter dois campos intercalados, um campo superior e um inferior, onde o campo superior contém a filas de números pares de pixéis, e o campo inferior contém a filas de número ímpares, iniciando na segunda linha da imagem 140. Neste caso, o macrobloco superior da região 144, se refere aos valores de pixel das linhas

do campo superior dentro da região 144, enquanto o macrobloco inferior da região 144 se refere ao conteúdo das linhas remanescentes. Dessa maneira, neste caso, ambos os macroblocos pressupões espacialmente, substancialmente a área total da região 144 com uma resolução de verticalidade reduzida. No caso de modo de moldura, o macrobloco superior é definido para abranger espacialmente a metade superior das filas dentro da região 144, enquanto o macrobloco inferior compreende as amostras de imagem remanescentes na região 144.

Conforme já observado acima, a unidade de transformação 108 executa uma transformação em âmbito de bloco do da saída do sinal residual pelo subtrator 106. Com relação a isto, a base de bloco para a transformação dentro da unidade de transformação 108 pode diferir do tamanho do macrobloco dos macroblocos 142. Em especial, cada um dos macroblocos 142 pode ser dividido em quatro, isto é, 2x2, blocos de transformada 146 ou 16, isto é, 4x4, blocos de transformada 148. Usando o exemplo mencionado acima para o tamanho do macrobloco de amostras de imagem de 16x16, a unidade de transformação 108 transformaria os macroblocos 142 da imagem 140 em âmbito de bloco, em blocos de tamanho de amostras de pixel 4x4 ou amostras de pixel 8x8. Dessa maneira, a unidade de transformação 108 envia, para certo macrobloco 142, vários blocos de transformada 146 e 148, respectivamente, isto é 16 blocos de coeficiente de transformada 4x4 ou 4 blocos de coeficiente de transformada 8x8 146.

Em 150 na Figura 5, um exemplo de um bloco de coeficiente de transformada 8x8 de um macrobloco codificador por moldura é ilustrado. Em especial, em 150, cada coeficiente de

transformada é designado e representado por um número de posição de exploração, estes números variando de 0 a 63. Conforme ilustrado pelos eixos 152, os coeficientes de transformação respectivos são associados com um componente de freqüência espacial diferente. Em especial, a freqüência associada com uma respectiva dos coeficientes de transformada aumenta em magnitude de um canto superior esquerdo para o canto inferior direito do bloco de transformada 150. A ordem de exploração definida pelas posições de exploração entre os coeficientes de transformada do bloco de transformada 150, explora os coeficientes de transformada a partir do canto superior esquerdo em ziguezague até o canto inferior direito, esta exploração em ziguezague sendo ilustrada por setas 154.

Apenas por uma questão de inteireza, é observado que a exploração entre os coeficientes de transformada pode ser diferentemente definida entre os coeficientes de transformada de uma bloco de transformada de um macrobloco codificado por campo. Por exemplo, conforme é mostrado em 156 na Figura 5, no caso de um macrobloco codificado por campo, a exploração de coeficiente de transformada 158 explora os coeficientes de transformada do canto superior esquerdo para o canto inferior direito em ziguezague com uma direção em ziguezague ou vaivém que é mais inclinada que a direção de ziguezague de 45° usada no caso do macrobloco codificado por moldura em 150. Em especial, uma exploração de coeficiente 158 explora os coeficientes de transformada na direção da coluna duas vezes mais rápido do que na direção da linha, de modo a considerar o fato de que os macroblocos codificados por campo abrangem amostras de imagem tendo um passo de coluna duas

vezes o passo horizontal ou linha. Dessa maneira, como é o caso com a exploração de coeficiente 154, a exploração de coeficiente 158 explora os coeficientes de transformada de uma maneira que a freqüência aumenta conforme o número de exploração de posição aumenta.

Em 150 e 158, exemplos para explorações de coeficiente de blocos de coeficiente de transformada 8x8 são mostrados. Entretanto, conforme já observado acima, blocos de transformada de tamanho menor, isto é, coeficientes de transformada 4x4, podem também existir. Para esses casos, explorações de posição respectivas são mostradas na Figura 5 em 160 e 162, respectivamente, com a exploração 164, no caso de 160 ser dedicada a macroblocos codificados por moldura, enquanto a exploração 166, ilustrada em 162, é dedicada para macroblocos codificados por campo.

Deve ser enfatizado que os exemplos específicos mostrados na Figura 5 com relação aos tamanhos e arranjos dos macroblocos e blocos de transformada, são de natureza ilustrativa apenas, e que variações diferentes são prontamente aplicáveis. Antes de iniciar com a descrição das figuras subseqüentes, é observado que a imagem 140 pode ser subdividida, em uma base de macrobloco, em vários pedaços 168. Um pedaço 168 é exemplificativamente mostrado na Figura 5. Um pedaço 168 é uma seqüência de macroblocos 142. A imagem 140 pode ser dividida em um ou em vários pedaços 168.

Após ter descrito a subdivisão de uma imagem em regiões de par de macrobloco, macroblocos e blocos de transformada, assim como pedaços, respectivamente, a

funcionalidade da unidade de distribuição 126 e a unidade de formação 128 são descritos a seguir em maiores detalhes. Como pode ser visto da Figura 5, a ordem de exploração definida entre os coeficientes de transformada permite aos coeficientes de transformada arranjados bidimensionalmente serem ordenados em uma seqüência linear de coeficientes de transformada com conteúdos de freqüência aumentando monotonamente, aos quais eles se referem. A unidade de distribuição 126 opera para distribuir os coeficientes de transformada de vários macroblocos 142 para camadas de qualidade diferentes, isto é, qualquer camada de ordem zero associada com uma corrente de dados 28 e com as camadas de refinamento 30. Em especial, a camada de distribuição 126 tenta distribuir os coeficientes de transformada para as correntes de dados 28 e 30 de maneira tal que com o número crescente de camadas de contribuição a partir de zero ou a camada 28 até a camada de refinamento da maior qualidade 30, a qualidade SNR do vídeo reconstrutivel das correntes de dados respectivas aumenta. Em geral, isto conduzirá a uma distribuição onde os coeficientes de transformada de menor freqüência correspondendo às menores posições de exploração são distribuídos para camadas de qualidade inferior, enquanto coeficientes de transformada de maior freqüência são distribuídos para camadas de qualidade superior. Por outro lado, a unidade de distribuição 126 tenderá a distribuir os coeficientes de transformada com valores de coeficiente de transformada maior para camadas de qualidade inferior e coeficientes de transformada com valores de coeficiente de transformada menor ou energias para camadas de qualidade superior. A distribuição formada pela unidade de distribuição 126 pode ser

executada de maneira que cada um dos coeficientes de transformada seja distribuído em uma camada única. Entretanto, é também possível que a distribuição executada pela unidade de distribuição 126 seja executada de maneira que a quantidade de um coeficiente de transformada pode também ser distribuída para diferentes camadas de qualidade em partes, de modo que as partes distribuídas sejam somadas ao valor de coeficiente de transformada. Detalhes de diferentes possibilidades para a distribuição executada pela unidade de distribuição 126 será descrita a seguir com relação às Figuras 6a-g. A unidade de formação 128 usa a distribuição resultante da unidade de distribuição 126, de modo a formar respectivas correntes de subdados 28 e 30. Como já foi observado acima, a corrente de subdados 28 forma a corrente de subdados de refinamento de camada de qualidade mais inferior e contém, por exemplo, os dados de movimento inseridos na entrada 124. Esta corrente de subdados de ordem zero 128 pode também ser provida com uma primeira porção distribuída dos valores de coeficiente de transformada. Dessa maneira, a corrente de subdados 28 permite um refinamento da corrente de dados de camada de qualidade base 26 para uma qualidade superior - no caso da Figura 1 para uma qualidade espacial superior - mas uma intensificação de qualidade SNR adicional pode ser obtida pelo acompanhamento da corrente de subdados 28 com qualquer das correntes de subdados de refinamento de qualidade superior adicionais 30. O número destas correntes de subdados de qualidade de refinamento 30 é N, onde N pode ser um ou mais que um. Os coeficientes de transformada são aqui - por exemplo, na ordem de importância de aumento para a qualidade SNR - "distribuídos" para estas correntes de subdados 28 e 30.

A Figura 6a mostra um exemplo para uma distribuição dos primeiros valores de coeficiente de transformada 26 de um bloco de transformada 8x8. Em especial, a Figura 6a mostra uma tabela onde a primeira linha da tabela lista as respectivas posições de exploração de acordo com a ordem de exploração 154 e 158, respectivamente (Figura 5). Pode ser visto que as posições de exploração mostradas se estendem, exemplificativamente, de 0 a 25. As três linhas seguintes mostram os valores de contribuição correspondentes incorporados nas respectivas correntes de subdados 28 e 30, respectivamente, para os valores de coeficiente de transformada individuais. Em especial, a segunda linha corresponde a, por exemplo, a corrente de subdados de ordem zero 28, enquanto a penúltima linha pertence à próxima camada de refinamento superior 30 e a última linha se refere a próxima corrente de dados de refinamento de qualidade par. De acordo com o exemplo da Figura 6a, um "122" é codificado nas correntes de subdados 128 para o componente DC, isto é, valor de coeficiente de transformada pertencendo à posição de exploração 0. Os valores de contribuição para este coeficiente de transformada tendo posição de exploração 0 dentro das duas correntes de subdados 30 a seguir, são definidos como zero conforme indicado pelo espalhamento das respectivas entradas da tabela. Desta maneira, de acordo com o exemplo da Figura 6a, a corrente de subdados da camada de intensificação de ordem zero 28 compreende um valor de distribuição para cada um dos valores de coeficiente de transformada. Entretanto, dentro do bloco de transformada da Figura 6a, meramente os valores de coeficiente de transformada de posições de exploração de 0 a 6, 8 e 9 pertencem

às camadas de qualidade de ordem zero. Valores de coeficiente de transformada adicionais são definidos como zero. Deve ser enfatizado, que em outros blocos de transformada, os valores de coeficiente de transformada pertencendo à camada de qualidade de ordem zero podem pertencer a outras posições de exploração. Similarmente, os valores de coeficiente de transformada de posições de exploração 7, 10 a 12, 15 a 18 e 21, pertencem à próxima camada de qualidade superior. Os valores de coeficiente de transformada remanescentes são definidos como zero. Os valores de coeficientes remanescentes das posições de exploração remanescentes são incluídos na próxima corrente de subdados de camada de qualidade superior. Como pode ser visto, pode ser possível que certo valor de coeficiente de transformada seja realmente zero. No exemplo da Figura 6a, este é o caso para a posição de exploração 23. Os valores de contribuição correspondentes dentro das camadas de qualidade precedentes são definidos como zero e o valor de coeficiente de transformada para a posição de exploração 23 na última camada de qualidade (última linha) para a posição de exploração 23 é, ela mesma, zero.

Dessa maneira, para cada uma das posições de exploração, os valores de contribuição incluídos nas várias correntes de sub-bits de camadas de qualidade 28 e 30, são somadas ao valores de coeficiente de transformada reais de modo que, no lado do decodificador, o bloco de transformada real pode ser reconstruído pela soma dos valores de contribuição para as posições de exploração individuais das diferentes camadas de qualidade.

De acordo com a configuração da Figura 6a, cada

uma das correntes de subdados 28 e 30 compreende um valor de contribuição para todos os coeficientes de transformada e para todas as posições de exploração, respectivamente. Entretanto, este não é necessariamente o caso. Primeiramente, conforme já mencionado acima, não é necessário que a corrente de subdados de ordem zero 28 contenha qualquer coeficiente de transformada ou valor de contribuição. Assim no último caso, as últimas três linhas da tabela da Figura 6a poderiam ser vistas como pertencendo às primeiras correntes de subdados de camada de refinamento 30 com a corrente de subdados de ordem zero 28 meramente compreendendo a informação de movimento da entrada 124.

Além do mais, é observado que os valores de contribuição da Figura 6a definidos como zero e valores de coeficiente de transformada reais sendo realmente zero foram distinguidos pelo uso de entradas de tabela de dispersão meramente por uma questão de fácil compreensão da funcionalidade da unidade de informação 128. Entretanto, as correntes de subdados 28 e 30 podem ser construídas de modo que a distinção mencionada acima entre valores de contribuição que foram definidos como zero e valores de contribuição que são naturalmente zero é transparente para o decodificador. Para ser mais preciso, alguns dos respectivos valores de contribuição para posições de exploração respectivas, isto é, os números da segunda até a quarta linha abaixo de uma respectiva posição de exploração na primeira linha da Figura 6a, revela o valor de coeficiente de transformada independente dos valores de contribuição individuais na soma terem sido definidos como zero ou serem naturalmente zero.

Na configuração da Figura 6a, a unidade de

formação 128 codificada em uma corrente de subdados respectiva da corrente de subdados 28 e 30, respectivamente, um valor de contribuição para cada uma das posições de exploração. Isto não é necessário. De acordo com a configuração da Figura 6b, por exemplo, as correntes de subdados da camada de qualidade consecutiva compreende meramente aqueles valores de coeficiente de transformada que pertencem à respectiva camada de qualidade.

A ordem, na qual os valores de contribuição e valores de coeficiente de transformada são codificados nas correntes de subdados 28 e 30, respectivamente, podem variar nas configurações da Figura 6a e Figura 6b, respectivamente. Por exemplo, as correntes de subdados 28 e 30 podem ser correntes de dados em pacotes, onde cada pacote corresponde a um pedaço 168. Dentro de um pedaço 168, os valores de coeficiente de transformada podem ser codificados nos respectivos pacotes no modo de bloco. Isto é, um ordem de exploração pode ser definida entre os macroblocos 142 dentro de um pedaço 168 com os valores de coeficiente de transformada para um macrobloco predeterminado 142 sendo completamente codificados no respectivo pacote antes do primeiro valor de coeficiente de transformada de um macrobloco seguindo uma ordem de exploração de macrobloco. Dentro de cada macrobloco, uma ordem de exploração pode ser definida entre os respectivos blocos de transformada 146 e 148, respectivamente, dentro do respectivo macrobloco. Novamente, os valores de coeficiente de transformada podem ser codificados em uma corrente respectiva das correntes de subdados 28 e 30, respectivamente pela unidade de formação 128, de modo que os valores de coeficiente de transformada de um bloco respectivo dos blocos de transformada

sejam todos codificados na respectiva corrente de subdados antes do primeiro valor de coeficiente de transformada de um próximo bloco de transformada ser codificado no mesmo. Dentro de cada bloco de transformada, uma codificação dos valores de coeficiente de transformada e valores de contribuição, respectivamente, podem ser conduzidos de uma maneira explicada a seguir com relação à Figura 8 ou 9.

De acordo com as configurações das Figuras 6a e 6b, os valores de coeficiente de transformada dos diferentes blocos de transformada do pedaço 168, que pertencem a uma camada respectiva das camadas de qualidade, se estendem sobre uma porção diferente da ordem de exploração. Para ser mais preciso, embora no bloco de transformada especifico exemplificativamente mostrado nas Figuras 6a e 6b, posições de exploração de 0 a 6, 8 e 9 pertencem à camada de qualidade de ordem zero, em um outro bloco de transformada, o conjunto posições de exploração pertencendo a esta camada pode ser diferente. De acordo com a configuração da Figura 6c, entretanto, a unidade de distribuição 126 distribui os valores de coeficiente de transformada dos diferentes blocos de transformada dentro de um pedaço 168, de modo que para todos os blocos de transformada, os valores de coeficiente de transformada do mesmo conjunto de posições de exploração pertencem à mesma camada de qualidade. Por exemplo, na Figura 6c, os valores de coeficiente de transformada das posições de exploração de 0 a 11 pertencem à corrente de subdados de ordem zero 28 com isto sendo verdadeiro para todos os blocos de transformada dentro do pedaço 168.

De acordo com a configuração da Figura 6c,

adicionalmente, os valores de coeficiente de transformada que pertencem a uma camada específica dentre as camadas de qualidade se estendem sobre uma seqüência contínua de posições de exploração consecutivas. Este, entretanto, não necessita ser o caso. Em especial, valores de coeficiente de transformada que pertencem a uma posição de exploração entre a primeira e a última posição de exploração pertencendo a uma camada de qualidade específica podem pertencer a uma camada dentre as outras camadas de qualidade tal como mostrado na Figura 6b. Entretanto, no caso da configuração da Figura 6c, é possível indicar as posições de exploração incorporadas em qualquer uma das correntes de subdados da camada de qualidade 28 e 30, respectivamente, meramente pelo uso de dois elementos de sintaxe, um indicando a primeira posição de exploração da respectiva camada de qualidade, isto é, scan_idx_start e o outro indicando a última posição de exploração para a respectiva camada de qualidade, isto é, scan_idx_end.

A reserva de um conjunto específico de posições de exploração para uma camada respectiva dentre as camadas de qualidade por um lado e a distribuição dependente da importância da qualidade dos coeficientes de transformada para as camadas de qualidade individuais por outro lado, pode ser misturada conforme mostrado na configuração a seguir. Por exemplo, a Figura 6d mostra uma configuração onde a unidade de distribuição 126 distribuiu os coeficientes de transformada sobre as camadas de qualidade conforme foi mostrado com relação à Figura 6a. Esta distribuição difere de bloco de transformada para bloco de transformada. Entretanto, por outro lado, cada uma das camadas de qualidade é designada para uma porção específica das posições de exploração em

comum para todos os blocos de transformada. Por exemplo, a camada de menor qualidade é designada para o conjunto total de posições de exploração da posição de exploração 0 até a posição de exploração 63. Dessa maneira, para cada bloco de transformada, a camada de menor qualidade compreende 64 valores de contribuição. A próxima corrente de subdados de camada de qualidade superior compreende valores de contribuição ou de coeficiente de transformada para todos os blocos de transformada em uma específica faixa de posição de exploração que se estende da posição de exploração 6 até a 63. A faixa de posição de exploração da próxima camada de qualidade se estende da posição de exploração 13 até a 63. Novamente, o decodificador não necessita saber se um valor específico dentre os valores de contribuição é um valor de contribuição que foi definido como 0 (entrada de espalhamento) ou está realmente indicando um valor de coeficiente de transformada de 0 ou valor de coeficiente de transformada insignificante. Entretanto, ele necessita saber o elemento de sintaxe scan_idx_start que indica para o pedaço respectivo 168 do qual a posição de exploração no coeficiente de transformada ou valores de contribuição contidos na respectiva corrente de subdados deve ser usado. Para ser mais preciso, na configuração da Figura 6d, por exemplo, a corrente de subdados correspondendo à penúltima linha compreende, para um bloco de transformada individual 58, coeficiente de transformada ou valores de contribuição. O primeiro, no caso do bloco de transformada da Figura 6d, é 0, enquanto o segundo é 22. Pelo uso do elemento de sintaxe scan_idx_start em um lado do decodificador, é sabido que o primeiro valor de coeficiente de transformada da respectiva camada

de qualidade corresponde à posição de exploração 6, enquanto os valores de coeficiente de transformada remanescentes desta camada de qualidade se referem às posições de exploração seguintes. Similar às configurações da Figura 6d, a Figura 6e mostra uma configuração onde um elemento de sintaxe scan_idx_end indica para as correntes de subdados individuais a última posição de exploração até a qual a respectiva corrente de subdados da camada de qualidade compreende subcoeficientes ou valores de contribuição.

Uma combinação das configurações das Figuras 6d e 6e é mostrada na Figura 6f. De acordo com esta configuração, o respectivo conjunto de posições de exploração que pertence a uma camada especifica dentre as camadas de qualidade se estende a partir de uma primeira posição de exploração indicada por um elemento de sintaxe scan_idx_start até uma última posição de exploração indicada pelo elemento de sintaxe last_idx_end. Por exemplo, na camada de qualidade correspondente à penúltima linha, o respectivo conjunto de posição de exploração se estende a partir da posição de exploração 6 até a posição de exploração 21. Finalmente, a configuração da Figura 6g mostra que o uso do elemento de sintaxe scan_idx_start e/ou scan_idx_end pode ser combinado com o foco da configuração da Figura 6c de acordo com o qual a distribuição dos valores de coeficiente de transformação individuais do bloco de transformada diferente dentro de um pedaço 168 é comum aos blocos de transformada. Consequentemente, de acordo com a configuração da Figura 6g, dentro de uma camada especifica dentre as camadas de qualidade, todos os valores de coeficiente de transformada dentro de scan_idx_start até

scan_idx_end são distribuídos para a respectiva camada de qualidade. Portanto, diferindo da configuração da Figura 6f, na configuração da Figura 6g, todos os valores de coeficiente de transformada dentro da posição de exploração 6 até a posição de exploração 21 são designados para a camada de qualidade correspondendo à penúltima linha na Figura 6g. Diferindo da mesma, na configuração da Figura 6f, vários valores dentre os valores de contribuição dentro desta faixa de exploração de posição de 6 a 21 podem ser definidos como 0, onde a distribuição dos valores de coeficiente de transformada que foram definidos como 0 e valores de coeficiente de transformada que não foram definidos como 0 dentro desta faixa de posição de exploração de 6 a 21, pode ser diferente de qualquer um dos outros blocos de transformada dentro do pedaço atual.

A seguir, a cooperação entre codificador híbrido 42, unidade de codificação de camada 44, unidade de distribuição 126 e unidade de formação 128 é descrita ilustrativamente com relação à Figura 7 que mostra um exemplo para a estrutura das correntes de subdados 28 e 30, respectivamente. De acordo com a configuração da Figura 7, a unidade de formação 28 é designada de modo que as correntes de subdados individuais 28 e 30, respectivamente, são constituídas em pacotes, isto é, elas compreendem um ou mais pacotes. Em especial, a unidade de formação 128 pode ser designada para gerar um pacote para cada pedaço 168 dentro de uma imagem 140 dentro de cada corrente de sub-bit 28 e 30, respectivamente. Como é mostrado na Figura 7, um pacote pode compreender um cabeçalho de pedaço 170 por um lado e dados residuais 172 por outro lado, exceto corrente de sub-bit 28 que

opcionalmente compreende meramente o cabeçalho de pedaço dentro de cada um dos pacotes.

Com relação à descrição dos dados residuais 172, isto é dados residuais #1, dados residuais #2, ..., dados residuais #N, é feito referência à descrição acima com relação às Figuras 6a a 6g, onde, por exemplo, a segunda até quarta linhas nestas tabelas correspondem aos dados residuais #1, dados residuais #2 e dados residuais #3, por exemplo. Em ainda outras palavras, dados residuais 172 indicados na Figura 7 incluem os valores de coeficiente de transformada discutidos nas Figuras 6a a 6g, a distribuição dos quais entre as respectivas correntes de subdados 28 e 30 não é novamente descrita aqui. Entretanto, a Figura 7 mostra elementos de sintaxe adicionais contidos no cabeçalho do pedaço 170 e os dados residuais 172 que originam-se de codificador híbrido 42. Conforme descrito acima, o codificador híbrido 42 comuta, em uma base de macrobloco entre vários modos de predição inter-camadas, de modo a se basear na informação de movimento da camada base, ou gerar nova informação de movimento para um respectivo bloco de movimento da camada de refinamento superior com codificação de predição da informação de movimento como um residual para a informação de movimento da camada base, ou com codificação desta informação de movimento nova. Dessa maneira, conforme indicado na Figura 7, os dados residuais 172 podem compreender para cada macrobloco, elementos de sintaxe indicando parâmetros de movimento, modos de macrobloco tais como codificados por campo ou moldura, ou um modo de inferência indicando a reutilização dos parâmetros de movimento da camada base com o respectivo macrobloco. Isto é especificamente verdadeiro para a

corrente de subdados zero ou para 28. Entretanto, esta informação de movimento não é novamente refinada nas camadas de refinamento subseqüentes e as correntes de subdados de qualidades superiores seguintes 30_1 a 30_N, e portanto, a unidade de formação 128 é designada para deixar estes elementos de sintaxe referentes a macrobloco com relação a modos de macrobloco, parâmetros de movimento e indicação de modo de inferência nos dados residuais destas correntes de subdados 30_1 a 30_N distantes ou para definir os elementos de sintaxe nestas correntes de subdados 30_1 a 30_N para serem tanto iguais aos modos de macrobloco e parâmetros de movimento para o respectivo macrobloco contido na corrente de subdados 28 quanto para indicar o modo de inferência para o respectivo macrobloco, de modo a indicar que a mesma definição deve ser usada na respectiva camada de refinamento. De acordo com a configuração da presente invenção, todos os dados residuais 172 dentro das várias correntes de subdados 28 e 30 e 30_1 a 30_N são passadas usando a mesma estrutura de sintaxe de modo que também os dados residuais dentro das correntes de subdados de refinamento 30_1 a 30_N compreendam informação definida com relação a macrobloco no modo de macrobloco, parâmetro de movimento e/ou ativação/desativação de modo de inferência.

Como também pode ser derivado da Figura 7, a unidade de formação 128 pode ser designada para prover um cabeçalho de pedaço 170 com o elemento de sintaxe scan_idx_star e/ou scan_idx_end. Alternativamente, os dados de cabeçalho de pedaço 170 podem compreender outros elementos de sintaxe definindo para cada pedaço ou pacote individual, um conjunto de posições de exploração, os dados residuais correspondendo aos respectivos

dados de cabeçalho de pedaço relacionados. Conforme já foi indicado acima, os dados de cabeçalho de pedaço de pacotes da corrente de subdados 28 pode não compreender estes elementos de sintaxe referentes à definição de posições de exploração específicas a camada no caso da corrente de subdados 28 não compreender qualquer dado residual, mas meramente modos de macrobloco e/ou parâmetros de movimento e indicações de modo de inferência, respectivamente. Adicionalmente, conforme já observado acima, os dados de cabeçalho de pedaço 170 podem compreender meramente um dentre scan_idx_start e scan_idx_end. Finalmente, scan_idx_start e/ou scan_idx_end podem ser providos uma vez por categoria de tamanho de bloco de transformada, isto é, 4x4 e 8x8, ou apenas uma vez para cada pedaço/imagem/corrente de subdados comumente para todas as categorias de tamanho de bloco de transformada, com as respectivas medidas sendo tomadas para transferir scan_idx_start e scan_idx_end para outros tamanhos de bloco como será descrito a seguir.

Adicionalmente, os dados de cabeçalho de pedaço pode compreender um elemento de sintaxe indicando o nivel de qualidade. Para este fim, a unidade de formação 128 pode ser designada de modo que o elemento de sintaxe ou indicador de qualidade meramente distinga entre o nivel de qualidade de ordem zero 28 por um lado e as camadas de refinamento 30_1 a 30_N por outro lado. Alternativamente, o indicador de qualidade pode distinguir todas as camadas de qualidade entre as camadas de refinamento 28 e 30_1 a 30_N. Nos dois últimos casos, o indicador de qualidade permitiria a omissão de quaisquer modos de macrobloco definidos com base em macrobloco, parâmetros de movimento e/ou modos de

inferência dentro dos pacotes das correntes de subdados 30_1 a 30_N, visto que neste caso, no lado do decodificador, é sabido que estas correntes de subdados de camadas de refinamento 30_1 a 30_N meramente refinam os coeficientes de transformada com o uso de modos de macroblocos, parâmetros de movimento e modos de inferência da corrente de subdados de modo zero 28.

Embora não descrito em maiores detalhes acima, a unidade de formação 28 ode ser designada para codificar por entropia os pacotes dentro das correntes de subdados 28 e 30_1 a 30_N. Dentro desta configuração, as figuras 8 e 9 mostram exemplos possíveis para codificação dos coeficientes de transformada dentro dos dados residuais pertencentes a um bloco de transformada de acordo com duas configurações. A Figura 8 mostra um pseudocódigo de um primeiro exemplo para uma possível codificação dos coeficientes de transformada dentro de um bloco de transformada em qualquer um dos dados residuais 172. Imagine que o exemplo a seguir se aplica:

Posição de Exploração	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Número de Coeficiente	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Nível de coeficiente de transformada	7	6	-2	0	-1	0	0	1	0	0	0	0	0	0	0	0

Baseado neste exemplo, a seguir, o pseudocódigo da Figura 8 é explicado mostrando a maneira na qual a unidade de formação 128 pode codificar os dados residuais dentro de um dos blocos de transformada em qualquer uma das correntes de subdados.

De modo a transportar a informação dos coeficientes de transformada, de acordo com a Figura 8, primeiramente um parâmetro coeff_token é provido em 240. O

parâmetro coeff_token é uma palavra de código indicando o número de coeficientes diferentes de zero, isto é total_coeff(coeff_token), e o número de coeficientes de transformada na operação dos coeficientes de transformada tendo um valor absoluto igual a um no final da seqüência de coeficientes de transformada diferentes de zero, isto é, trailing_ones(coeff_token). Em nosso exemplo, total_coeff(coeff_token) é 5 (números de coeficiente de transformada 0, 1, 2, 4, e 7) e trailing_ones(coeff_token) é 2 (número de coeficiente de transformada 4 e 7). Assim, provendo o parâmetro coeff_tokem 240, as posições dos coeficientes de transformada significativos foram determinadas até a extensão de que não mais que total_coeff(coeff_token) coeficientes de transformada diferentes de zero existem.

Então, os valores dos níveis dos coeficientes de transformada diferentes de zero são providos. Isto é feito na ordem inversa de exploração. Para ser mais específico, primeiramente é verificado se o número total de coeficientes de transformada diferentes de zero é maior que zero 242. Este é o caso no exemplo acima, visto que total_coeff(coeff_token) é 5.

Então, os coeficientes diferentes de transformada são passados em uma ordem de exploração inversa 244. A ordem de exploração inversa não é ainda óbvia apenas da visão da incrementação do parâmetro de contagem i++ na "for-loop" 244, mas se tornará claro a partir da avaliação a seguir. Enquanto passando através destes coeficientes diferentes de transformada na ordem de exploração inversa, para o primeiro destes coeficientes de transformada diferentes de zero, apenas seu sinal de coeficiente

de transformada é provido 248. Isto é feito para o primeiro número de trailing_ones(coeff_token) dos coeficientes de transformada diferentes de zero quando passando através deles em uma ordem de exploração inversa, visto que para estes coeficientes de transformada já é sabido que o valor absoluto destes coeficientes de transformada é um (comparar com a definição acima de trailing_ones(coeff_token)). Os sinais de coeficiente assim providos são usados para armazenar temporariamente no nível de coeficientes de vetor auxiliar [i] para o nível de coeficiente de transformada dos níveis de coeficiente de transformada diferente de zero tendo valor absoluto de 1, onde i é uma numeração dos coeficientes de transformada diferentes de zero quando explorados na ordem de exploração inversa (250). Em nosso exemplo, após as duas primeira rodadas para o "for-loop" 244, nível[0] = 1 e nível[1] = -1 é obtido.

A seguir, os níveis de coeficiente coeff_level para os coeficientes de transformada diferentes de zero remanescentes são providos (252) na ordem de exploração inversa e temporariamente armazenados no nível[i] de coeficientes de vetor auxiliar (254). As rodadas "for-loop" remanescentes resultam no nível[2] = -2, nível[3] = 6 e nível[4] = 7.

Agora, de modo a tornar a determinação das posições dos coeficientes de transformada significativos única, dois parâmetros adicionais denominados total_zeros e run_before são providos a menos que total_coeff(coeff_token) já seja igual ao número máximo de coeficientes de transformada em um bloco de transformada, isto é, seja igual ao maxNumCoeff. Para ser mais específico, é verificado se o total_coeff(coeff_token) é igual ao

maxNumCoeff (256). Se este não for o caso, o parâmetro total_zeros é provido (258) e um parâmetro auxiliar zerosLeft é inicializado para o valor de total_zero (260). O parâmetro total_zeros especifica o número de zeros entre o último coeficiente diferente de zero na ordem de exploração e o início da exploração. No exemplo acima, total_zeros é 3 (números de coeficiente 3,5, e 6). Assim, zeroLeft é inicializado em 3.

Para cada um dos coeficientes de transformada diferentes de zero exceto o último com relação à ordem de exploração inversa (número de coeficiente 0), iniciando com o último coeficiente de transformada diferente de zero (número de coeficiente 7) com relação à ordem de exploração (62), um parâmetro run_before é provido (64) indicando o comprimento da operação de coeficientes de transformada de nível zero arranjados diretamente na frente do respectivo coeficiente de transformada diferente de zero quando visto na ordem de exploração. Por exemplo, para i sendo igual a zero, o último coeficiente de transformada diferente de zero com relação à ordem de exploração é o coeficiente de transformada diferente de zero em questão. Em nosso exemplo, este é coeficiente de transformada tendo o número 7 e tendo o nível 1. A operação de zeros na frente deste coeficiente de transformada tem um comprimento de 2, isto é, coeficientes de transformada 5 e 6. Dessa maneira, em nosso exemplo, o primeiro parâmetro run_before é 2. Este parâmetro é temporariamente armazenado na operação de coeficiente de vetor auxiliar [0] (266). Isto é repetido na ordem de exploração inversa para operação[i], com i sendo a contagem dos coeficientes de transformada diferentes de zero quando explorados na ordem de exploração inversa.

Diminuindo o parâmetro auxiliar zerosLeft do parâmetro run_before em cada rodada da "for-loop" (261) é determinado para cada rodada quantos coeficientes de transformada de nível zero são deixados. Se zerosLeft for zero, nenhum parâmetro run_before é provido adicionalmente (270) e os coeficientes remanescentes da operação do vetor são definidos como zero (272). Em qualquer caso, nenhum parâmetro run_before é provido para o último coeficiente de transformada diferente de zero quando passado na ordem de exploração inversa, isto é, diferente do parâmetro run_before para o primeiro coeficiente de transformada diferente de zero com relação à ordem de exploração. Este parâmetro é deduzido do número de coeficientes de transformada de nível zero deixado, conforme indicado pelo parâmetro auxiliar zerosLeft (274). Em nosso exemplo, os coeficientes de vetor auxiliar da operação do vetor são operação[0] = 2, operação[[1] = 1, operação[2] = 0, operação[3] =0, e operação[4] = 0.

Ao final, no "for-loop" indicado em 276, os valores dos níveis de coeficiente de transformada conforme armazenados no nível de vetor auxiliar são designados para suas posições pela cópia dos valores dos coeficientes do nível de vetor para a respectiva posição no arranjo unidimensional coeffLevel. Para ser mais específico, na primeira rodada do "for-loop" 276, i = 4 e coeffNum que foi inicializado em 0 (278) é incrementado pela operação[4] + 1 = 0 + 1 = 1 resultando no coeffNum = 0 e coeffLevel[0] sendo designado ao valor de nível]4] = 7. Isto é repetido para o próximo nível[3] de coeficientes de vetor auxiliar até o nível [0]. Visto que as posições remanescentes do arranjo coeffLevel foram inicializadas para o valor de zero (280) todos os

coeficientes de transformada foram codificados.

Os elementos de sintaxe expressos em negrito na Figura 8 podem ser codificados na respectiva corrente de subdados por meio de codificação de comprimento variável, por exemplo.

A Figura 9 mostra um outro exemplo para codificação de um bloco de transformada. Neste exemplo, a ordem de exploração é manifestada em "i++" dentro da "while-loop" 310 indicando que contagem de parâmetros i é incrementada por interação "while-loop".

Para cada coeficiente na ordem de exploração, um símbolo de um bit significant_coeff_flag é provido (312). Se o símbolo significant_coeff_flag for 1 (314), isto é, se um coeficiente diferente de zero existir nesta posição de exploração i, um símbolo de um bit adicional last_significant_coeff_flag é provido (316). Este símbolo indica se um coeficiente significativo atual é o último dentro do bloco ou se coeficientes significativos adicionais seguem na ordem de exploração. Dessa maneira, se o símbolo last_significant_coeff_flag for um (318), isto indica que o número de coeficientes, isto é, numCoeff, é i+1 (320) e os níveis dos coeficientes de transformada subseqüentes podem ser deduzidos para serm zero (322). Até aqui, os elementos de sintaxe last_significant_coeff_flag e significant_coeff_flag podem ser vistos como um mapa de significância. Então, para o último coeficiente de transformada na ordem de exploração, o valor absoluto do nível menos 1, isto é, coeff_abs_level_minus1, e seu sinal, isto é, coeff_sign_flag, é provido (324), portanto indicando o nível do coeficiente de transformada deste último coeficiente de transformada significativo (326). Estas etapas 324,

326 são repetidas (328) para os coeficientes de transformada significativos remanescentes (330) na ordem de exploração inversa (332), onde a ordem de exploração inversa é manifestada em i--, isto é, i = i -1, indicando que a contagem do parâmetro i é decrescida por rodada na "for-loop". A análise dos elementos de sintaxe coeff_abs_level_minus1 inicia com derivação de uma binarização para os valores possíveis do elemento de sintaxe. O esquema de binarização pode ser um UEGO, isto é, um processo de binarização Exp-Golomb de ordem concatenada unary/zero-th. Dependendo das binarizações possíveis, o elemento de sintaxe respectivo pode ser codificado de forma aritmética binária bin por bin. Com relação a isto, um esquema de codificação aritmética binária adaptativo de contexto pode ser usado para uma parte de prefixo da binarização de coeff_abs_level_minus1 enquanto usando um processo de desvio de decodificação que não tem adaptação para uma parte de sufixo.

Por uma questão de inteireza, é observado que se torna claro na Figura 5 que o número de posições de exploração distinguíveis dentro dos blocos de transformada 8x8 é 64, enquanto o número de posições de exploração distinguíveis dentro dos blocos de transformada 4x4 é meramente 16. Consequentemente, os elementos de sintaxe mencionados acima scan_idx_start e scan_idx_end podem tanto ser definidos com uma precisão que permite uma distinção entre todas as 64 posições de exploração, ou meramente uma distinção entre 16 posições de exploração. No último caso, por exemplo, os elementos de sintaxe podem ser aplicados a cada quádruplo de coeficientes de transformada consecutivos dentro dos blocos de transformada 8x8. Para ser mais preciso, blocos de

transformada 8x8 podem ser codificados pelo uso de

residual_block(LumaLevel8x8,

4*scan_idx_start, 4*scan_ids-end +3, 64)

e no caso de blocos de transformada 4x4 pelo uso de

residual_block(LumaLevel4x4,

scan_idx_start, scan_idx_end, 16).

com residual_block sendo tanto residual_block_cavlc ou residual_block_cabac, e LumaLevel4x4 e LumaLevel8x8 indicando um arranjo de amostras luma do respectivo bloco de transformada 4x4 e 8x8, respectivamente. Como pode ser visto, scan_idx_start e scan_idx_end são definidos para discriminar entre 16 posições de exploração, de modo que eles indiquem a faixa de posições nos blocos 4x4 exatamente. Entretanto, nos blocos 8x8, a precisão destes elementos de sintaxe não é suficiente, de modo que neste blocos a faixa é ajustada de forma quádrupla.

Além do mais, blocos 8x8 de coeficientes de transformada podem também ser codificados pela divisão dos coeficientes 64 de um bloco 8x8 em quatro conjuntos de 16 coeficientes, por exemplo, pela colocação de cada quarto coeficiente em um conjunto "n-th" (número ordinal) iniciando com o coeficiente n, com n na faixa de 0 a 3, inclusive, e codificando cada conjunto de 16 coeficientes usando a sintaxe de bloco residual para blocos 4x4. No lado do codificador, estes quatro conjuntos de 16 coeficientes são recombinados para formar um conjunto de 64 coeficientes representando o bloco 8x8.

Após ter descrito configurações para um

codificador, um decodificador para decodificar a respectiva corrente de dados de qualidade escalonável é explicado com relação às Figuras 10 e 11. A Figura 10 mostra a construção geral de um decodificador 400. O decodificador 400 compreende um desmultiplexador 402 tendo uma entrada 404 para receber a corrente de bits escalonável 36. O desmultiplexador 402 desmultiplexa o sinal de entrada 36 nas correntes de dados 26 a 32. Para este fim, o desmultiplexador pode executar uma decodificação e/ou função de análise. Por exemplo, o desmultiplexador 402 pode decodificar as codificações do bloco de transformada da Figura 8 e 9. Adicionalmente, reveja as Figuras 6a-6g. Consequentemente, o desmultiplexador 402 pode usar informação de correntes de subdados precedentes de modo a, na análise de uma corrente de subdados atual, saber quantos valores de coeficiente de transformada ou valores de contribuição devem ser esperados para um bloco de transformada específico. As correntes de dados assim recuperadas são recebidas por uma unidade de decodificação 406 que, com base nestas correntes de dados, reconstrói o vídeo 18 e envia o respectivo vídeo reconstruído 408 em uma respectiva saída 410.

A estrutura interna da unidade de decodificação 406 é mostrada em maiores detalhes na Figura 11. Conforme mostrado aqui, a unidade de decodificação 406 compreende uma entrada de dados de movimento de camada base 412, uma entrada de dados residuais de camada base 414, uma entrada de dados de movimento de camada de refinamento de ordem zero 416, uma entrada de dados de coeficiente de transformada de refinamento de ordem zero de coeficiente de transformada opcional 418 e uma entrada 420 para as correntes de subdados 30. Conforme mostrado, as entradas 412 e 414

são para receber corrente de dados 26, enquanto as entradas 416 e 418 cooperam para receber a corrente de dados 28. Além disso, a unidade de decodificação 406 compreende uma saída de sinal de vídeo de reconstrução de qualidade inferior 422, uma saída de sinal de vídeo de reconstrução codificada inter-camadas de qualidade superior 424, e uma saída de sinal de vídeo de reconstrução codificado internamente 426, as últimas provendo a informação para um sinal de vídeo de qualidade superior.

Um combinador 428 tem entradas conectadas às entradas 418 e 420 e uma saída para enviar níveis de coeficiente de transformada para os blocos de transformada individuais conforme obtido pela coleta dos valores de contribuição correspondentes a partir das várias camadas de qualidade. A coleta pode envolver uma soma dos valores de contribuição para um coeficiente de transformada específico dentro de várias das correntes de dados 30 e 28. Entretanto, é também possível que o combinador 428 predefina todos os valores de coeficiente de transformada como zero e substitua qualquer um desses zeros meramente no caso de um valor de contribuição ser diferente de zero para a respectiva posição de exploração. Com esta medida, o combinador coleta informação nos coeficientes de transformada dos vários blocos de transformada. A associação dos valores de contribuição ou coeficiente de transformada dentro das camadas individuais pode envolver o combinador usando a informação de posição de exploração da camada atual tal como scan_idx_start e/ou scan_idx_end. Alternativamente, o combinador pode usar o conhecimento dos valores de coeficiente de transformada dentro dos blocos de transformada individuais recebidos até agora de camadas

de qualidade inferior ou SNR.

A saída dos blocos de transformada pelo combinador 428 são recebidas por um decodificador de predição residual 430 e um adicionador 432.

Entre o decodificador de predição residual 430 e a entrada 414, uma unidade de retro-transformação ou inversa 432 é conectada de modo a enviar dados residuais inversamente transformados para o decodificador de predição residual 430. O último usa os dados residuais transformados inversamente de modo a obter um preditor a ser adicionado aos coeficientes de transformada da saída dos blocos de transformada pelo combinador 428, eventualmente após executar uma amostragem ascendente ou uma outra adaptação de qualidade. Por outro lado, uma unidade de predição de movimento 434 é conectada entre a entrada 412 e uma entrada de um adicionador 436. Uma outra entrada do adicionador 436 é conectada à saída de uma unidade de retro-transformação 432. Com esta medida, a unidade de predição de movimento 434 usa os dados de movimento na entrada 412 para gerar um sinal de predição para a saída de sinal residual transformado inversamente pela unidade de retro-transformação 432. Um resultado do adicionador 436 na saída do adicionador 436 é um sinal de vídeo de camada base reconstruído. A saída do adicionador 436 é conectada à saída 432 assim como na entrada do decodificador de predição 432. O decodificador de predição 432 usa o sinal de camada base reconstruído como uma predição para as porções codificadas de intracamada da saída de conteúdo de vídeo pelo combinador 428, eventualmente pelo uso de uma amostragem ascendente. Por outro lado, a saída do adicionador 436 é também conectada a uma entrada

de unidades de predição de movimento 434, de modo a permitir que a unidade de predição de movimento 434 use os dados de movimento na entrada 412 para gerar um sinal de predição para a segunda entrada do adicionador 436, baseada nos sinais reconstruídos da corrente de dados de camada base. Os valores de coeficiente de transformada decodificados por meio de predição enviados pelo decodificador de predição residual 430 são retro-transformados pela unidade de retro-transformação 438. A saída da unidade de retro-transformação 438 resulta em um sinal de vídeo de dados residuais de qualidade superior. Este sinal de vídeo de dados residuais de qualidade superior é adicionado por um adicionador 440 com uma saída de sinal de vídeo de predição de movimento por uma unidade de predição de movimento 442. Na saída do adicionador 440, o sinal de vídeo de qualidade superior reconstruído resulta, o qual atinge a saída 424 assim como uma entrada adicional da unidade de predição de movimento 442. A unidade de predição de movimento 442 executa a predição de movimento com base na saída de sinal de vídeo reconstruído pelo adicionador 440, assim como na saída de informação de movimento por um decodificador de predição de parâmetro de movimento 444 que está conectado entre a entrada 416 e uma entrada respectiva de unidade de predição de movimento 442. O decodificador de predição de parâmetro de movimento 444 usa, em uma base seletiva de macrobloco, dados de movimento da entrada de dados de movimento da camada base 412 como um preditor, e dependente destes dados, envia os dados de movimento para a unidade de predição de movimento 442 usando, por exemplo, os vetores de movimento na entrada 416 como vetores de deslocamento para vetores de movimento na entrada 412.

As configurações descritas acima permitem um aumento na granularidade de codificação escalonável SNR em um nível de imagem/pedaço em comparação com codificação CGS/MGS conforme descrito na porção introdutória, mas sem o aumento significativo na complexidade que está presente na codificação FGS. Além do mais, visto que é considerado que a característica de FGS de que pacotes podem ser truncados não será usada amplamente, a adaptação da corrente de bits é possível pelo simples descarte do pacote.

As configurações descritas acima têm a idéia básica em comum, dividir os níveis de coeficiente de transformada de um pacote CGS/MGS tradicional como ele é atualmente especificado no rascunho SVC em subconjuntos, que são transmitidos em diferentes pacotes e diferentes camadas de refinamento SNR. Como um exemplo, as configurações descritas acima referentes à codificação CGS/MGS com uma camada base e uma camada de intensificação. Ao invés da camada de intensificação incluindo, para cada imagem, modos de macroblocos, modos de intra-predição, vetores de movimento, índices de imagem de referência, outros parâmetros de controle, assim como níveis de coeficiente de transformada para todos os macroblocos, de modo a aumentar a granularidade da codificação escalonável SNR, estes dados foram distribuídos sobre diferentes pedaços, diferentes pacotes, e diferentes camadas de intensificação. Na primeira camada de intensificação, os modos de macrobloco, parâmetro de movimento, outros parâmetros de controle, assim como, opcionalmente, um primeiro subconjunto de níveis de coeficiente de transformada são transmitidos. Na próxima camada de intensificação, os mesmos modos

de macrobloco e vetores de movimento são usados, mas um segundo subconjunto de níveis de coeficiente de transformada são codificados. Todos os coeficientes de transformada que foram transmitidos na primeira camada de intensificação podem ser definidos como zero na segunda e em todas as camadas de intensificação. Em todas as camadas de intensificação subseqüentes (terceira, etc..), os modos de macroblocos e parâmetros de movimento da primeira camada de intensificação são novamente usados, mas subconjuntos adicionais de níveis de coeficiente de transformada são codificados.

Deve ser observado que esta divisão não aumenta ou apenas aumenta muito ligeiramente a complexidade em comparação com a codificação CGS/MGS tradicional conforme especificado no rascunho SVC atual. Todas as intensificações SNR podem ser analisadas paralelamente, e os coeficientes de transformada não necessitam ser coletados de explorações diferentes sobre a imagem/pedaço. Isto significa, por exemplo, que um decodificador poderia analisar todos os coeficientes de transformada para um bloco de todas as intensificações SNR, e então ele pode aplicar a transformada inversa a este bloco sem armazenar os níveis de coeficiente de transformada em uma memória temporária. Quando todos os blocos de um macrobloco foram completamente analisados, a predição compensada por movimento pode ser aplicada e o sinal de reconstrução final para este macrobloco pode ser obtido. Deve ser observado que todos os elementos de sintaxe em um pedaço são transmitidos macrobloco por macrobloco, e dentro de um macrobloco, os valores de coeficiente de transformada são transmitidos de bloco de transformada para bloco de transformada.

É possível que um marcador seja codificado no nível de pedaço, que sinaliza se todos os modos de macrobloco e parâmetros de movimento são inferidos da camada base. Dada a sintaxe atual de pacotes CGS/MGS que significa especialmente que todos os elementos de sintaxe mb_skip_run e mb_skp_flag não são transmitidos mas inferidos como sendo iguais a 0, que todos os elementos de sintaxe mb_field_decoding_flag não são transmitidos mas inferidos como sendo iguais a seus valores nos macroblocos de camada base co-posicionados, e que todos os elementos de sintaxe base_mode_flag e residual_prediction_flag não são transmitidos, mas inferidos como sendo iguais a 1. Na primeira camada de intensificação SNR este marcador deveria usualmente ser definido como 0, visto que para esta intensificação deveria ser possível transmitir vetores de movimento que sejam diferentes da camada base de modo a melhorar a eficiência de codificação. Mas em todas as camadas de intensificação adicionais, este marcador é definido como igual a 1, visto que estas camadas de intensificação representam apenas um refinamento de níveis de coeficiente de transformada de posições de exploração que não foram codificados nas camadas de intensificação SNR anteriores. E ao definir este marcador como igual a 1, a eficiência de codificação pode ser melhorada para este caso, visto que nenhuma transmissão de elementos de sintaxe não requeridos é necessária e assim a taxa de bits associada é salvada.

Conforme descrito adicionalmente acima, a primeira posição de exploração x para os níveis de coeficiente de transformada nos vários blocos de transformada pode ser transmitida em um nível de pedaço, sem elementos de sintaxe

transmitidos em um nível de macrobloco para coeficientes de transformada com uma posição de exploração que seja menor que x. Adicionalmente à descrição acima onde a primeira posição de exploração é transmitida apenas para um tamanho de transformada específico e a primeira posição de exploração para outros tamanhos de transformada é inferida com base no valor transmitido, seria possível transmitir uma primeira posição de exploração para todos os tamanhos de transformada suportados.

Similarmente, a última posição de exploração y para os níveis de coeficiente de transformada nos vários blocos de transformada pode ser transmitida em um nível de pedaço, sem elementos de sintaxe transmitidos em um nível de macrobloco para coeficientes de transformada com uma posição de exploração que seja maior que y. Novamente, é possível tanto transmitir uma última posição de exploração para todos os tamanhos de transformada suportados, ou transmitir a última posição de exploração apenas para um tamanho de transformada específico e inferir a última posição de exploração para outros tamanhos de transformada com base no valor transmitido.

A primeira posição de exploração para cada bloco de transformada em uma camada de intensificação SNR pode, alternativamente, ser inferida com base nos coeficientes de transformada que foram transmitidos em uma camada de intensificação anterior. Esta regra de inferência pode ser independentemente aplicada a todos os blocos de transformada, e em cada bloco um primeiro coeficiente de transformada diferente pode ser derivado, por exemplo, pelo combinador 428.

Adicionalmente, uma combinação de sinalização e

inferência da primeira posição de exploração pode ser efetuada. Isto significa que a primeira posição de exploração pode basicamente ser inferida com base nos níveis de coeficiente de transformada já transmitidos em camadas de intensificação SNR anteriores, mas para isto é usado o conhecimento adicional de que a primeira posição de exploração não pode ser menor que um valor x, que é transmitido no cabeçalho do pedaço. Com este conceito é novamente possível ter um primeiro índice de exploração diferente em cada bloco de transformada, que pode ser escolhido de modo a maximizar a eficiência de codificação.

Como ainda uma alternativa adicional, a sinalização da primeira posição de exploração, a inferência da primeira posição de exploração, ou a combinação delas pode ser combinada com a sinalização da última posição de exploração.

Até aqui, a descrição acima permite um esquema possível permitindo escalonabilidade SNR na qual apenas subconjuntos de níveis de coeficiente de transformada são transmitidos em camadas de intensificação SNR diferentes, e este modo é sinalizado por um ou mais elementos de sintaxe de cabeçalho de pedaço, que especificam que modos de macrobloco e parâmetros de movimento são inferidos para todos os tipos de macrobloco e/ou que coeficientes de transformada para várias posições de exploração não estão presentes em um nível de bloco de transformada. Um elemento de sintaxe de nível de pedaço pode ser usado que sinaliza que os modos de macrobloco e parâmetros de movimento para todos os macroblocos são inferidos dos macroblocos de camada base co-posicionada. Especificamente, os mesmos modos de macrobloco e parâmetros de movimento podem ser usados, e os elementos de

sintaxe correspondentes podem não ser transmitidos em um nível de pedaço. A primeira posição de exploração x para todos os blocos de transformada pode ser sinalizada pelos elementos de sintaxe de cabeçalho de pedaço. No nível de macrobloco, nenhum elemento de sintaxe é transmitido para valores de coeficiente de transformada de posições de exploração menores que x. Alternativamente, a primeira posição de exploração para um bloco de transformada pode ser inferida com base nos níveis de coeficiente de transformada transmitidos da camada base. Uma combinação das últimas alternativas é também possível. Similarmente, a última posição de exploração y para todos os blocos de transformada pode ser sinalizada por elementos de sintaxe de cabeçalho de pedaço, onde, no nível de macrobloco, nenhum elemento de sintaxe é transmitido para valores de coeficiente de transformada de posições de exploração maiores que y.

Conforme observado acima, as configurações descritas detalhadamente das Figuras 1-11 podem ser variadas de várias maneiras. Por exemplo, embora as configurações acíma tenham sido exemplificadas com relação a um ambiente de duas camadas espaciais, as configurações acima são prontamente transferíveis para uma configuração com apenas uma camada de qualidade ou com mais que uma camada de qualidade, mas com as camadas de refinamento escalonável SNR N+1. Imagine, por exemplo, que a parte 12 na Figura 1 esteja faltando. Neste caso, codificador híbrido 42 age como um meio de codificação para codificar o sinal de vídeo 18 usando transformação na forma de bloco para obter blocos de transformada 146, 148 de valores de coeficiente de transformação para uma imagem 140 do sinal de vídeo, enquanto a unidade 44 age

como um meio para formar, para cada uma dentre uma pluralidade de camadas de qualidade, uma corrente de subdados de vídeo 30 ou 28 mais 30 contendo informação de faixa de exploração indicando um subconjunto das possíveis posições de exploração, e informação de coeficiente de transformada nos valores de coeficiente de transformação que pertencem ao subconjunto de possíveis posições de exploração. Nenhuma predição de inter-camadas estaria envolvida. Além do mais, o codificador 42 pode ser simplificado para executar predição sem movimento, mas meramente transformação na forma de bloco. Similarmente no caso de uma camada de qualidade, o desmultiplexador 402 agiria como um meio de análise para analisar as correntes de subdados de vídeo da pluralidade de camadas de qualidade para obter, para cada camada de qualidade, a informação de faixa de exploração e a informação de coeficiente de transformada, e o combinador 428 agiria como um meio para, usando a informação de faixa de exploração, para cada camada de qualidade, construir os blocos de transformada pela associação dos valores de coeficiente de transformação dos respectivos blocos de transformada das possíveis posições de exploração, com a unidade de retro-transformação 438 reconstruindo a imagem do sinal de vídeo por uma retro-transformação dos blocos de transformada.

Além do mais, a configuração na Figura 1 pode ser variada de uma maneira que o codificador da camada base 12 opere com a mesma resolução espacial e a mesma profundidade de bits que o codificador de camadas de intensificação 14. Neste caso, a configuração representa codificação escalonável SNR com uma camada base padrão 26 e várias camadas de intensificação 28, 30 que contém partições dos coeficientes de transformada.

Dependendo da implementação real, o esquema inventivo pode ser implementado em hardware ou software. Portanto, a presente invenção também se refere a um programa de computador, que pode ser armazenado em um meio legível por computador tal como um CD, disco ou qualquer outro armazenador de dados. A presente invenção é, portanto, também um programa de computador tendo um código de programa que, quando executado em um computador, executa o método inventivo em conexão com as figuras acima.

Além do mais, é observado que todas as etapas ou funções indicadas nos diagramas de fluxo poderiam ser implementadas pelos respectivos meios no codificador e que as implementações podem compreender sub-rotinas operando em uma CPU, partes de circuito de um ASIC ou similares.

REIVINDICAÇÕES

1. Aparelho para gerar uma corrente de dados de vídeo de qualidade escalonável (36), em que uma sequência unidimensional de valores de coeficiente de transformação é organizada em uma pluralidade de camadas de qualidade, caracterizado pelo fato de que compreende:

meios (42) para codificar um sinal de vídeo (18) usando transformação na forma de bloco para obter blocos de transformada (146, 148) de valores de coeficiente de transformada arranjados bidimensionalmente para uma imagem (140) de um sinal de vídeo, onde uma ordem de exploração predeterminada (154, 156, 164, 166) com possíveis posições de exploração ordena os valores de coeficiente de transformada na da dita sequencia unidimensional de valores de coeficiente de transformada; e

meios (44) para formar, para cada uma dentre a pluralidade de camadas de qualidade, uma corrente de subdados de vídeo (30; 28, 30) contendo informação de faixa de exploração que indica um subconjunto das possíveis posições de exploração, de modo que o subconjunto de cada uma dentre a pluralidade de camadas de qualidade compreende pelo menos uma possível posição de exploração não incluída no subconjunto de qualquer outra dentre a pluralidade de camadas de qualidade e uma dentre as possíveis posições de exploração é incluída em mais que um dos subconjuntos das camadas de qualidade, e informação de coeficiente de transformada nos valores de coeficiente de transformação pertencendo ao subconjunto de possíveis posições de exploração da respectiva camada de qualidade, contendo um

valor de contribuição por posição de exploração possível do subconjunto das possíveis posições de exploração da respectiva camada de qualidade, de modo que o valor de coeficiente de transformada da posição de exploração possível é derivável, com base em uma soma de valores de contribuição para a posição de exploração possível, de mais que um dos subconjuntos das camadas de qualidade.

2. Aparelho para reconstrução de um sinal de vídeo de uma corrente de dados de vídeo de qualidade escalonável (36) em que uma sequência unidimensional de valores de coeficiente de transformação é organizada em uma pluralidade de camadas de qualidade, e que compreende para cada pluralidade de camadas de qualidade, uma corrente de subdados de vídeo (30; 28 30), caracterizado pelo fato de que compreende:

meios (402) para analisar as correntes de subdados de vídeo da pluralidade de camadas de qualidade, para obter, para cada camada de qualidade, uma informação de faixa de exploração e informação de coeficiente de transformada nos valores de coeficiente de transformação arranjados bidimensionalmente de diferentes blocos de transformada, onde uma ordem de exploração predeterminada com possíveis posições de exploração ordena os valores de coeficiente de transformada da dita sequência unidimensional de valores de coeficiente de transformação, e a informação de faixa de exploração indica um subconjunto das possíveis posições de exploração;

meios (428) para construir, usando a informação de faixa de exploração para cada camada de qualidade, os blocos de transformada pela associação com os valores de coeficiente de

transformada dos respectivos blocos de transformada da informação de coeficiente de transformada para o subconjunto das possíveis posições de exploração; e

meios (438) para reconstruir uma imagem do sinal de vídeo por uma retro-transformação dos blocos de transformada, onde

os meios para análise são configurados de modo que a informação de coeficiente de transformada de mais que uma dentre as camadas de qualidade contem um valor de contribuição relacionado a um valor de coeficiente de transformada, e onde os meios para construção estão configurados para derivar o valor para o valor de coeficiente de transformada baseado em uma soma dos valores de contribuição em relação a um valor de coeficiente de transformada.

3. Aparelho, de acordo com a reivindicação 2, caracterizado pelo fato de que os meios (428) para construção são configurados para usar a informação de faixa de exploração como se a mesma indicasse uma primeira posição de exploração entre as possíveis posições de exploração dentro do subconjunto de possíveis posições de exploração na ordem de exploração predeterminada.

4. Aparelho, de acordo com a reivindicação 2 ou 3, caracterizado pelo fato de que os meios (428) para construção são configurados para usar a informação de faixa de exploração como se a mesma indicasse uma última posição de exploração entre as possíveis posições de exploração dentro do subconjunto de possíveis posições de exploração na ordem de exploração predeterminada.

5. Aparelho, de acordo com qualquer uma das reivindicações de 2 a 4, caracterizado pelo fato de que os meios para reconstrução são configurados para reconstruir o sinal de vídeo usando predição de movimento baseada em informação de movimento e por combinação de um resultado de predição de movimento com um residual de predição de movimento, com obtenção do residual de predição de movimento por uma transformação inversa em relação a bloco dos blocos de transformada de valores de coeficiente de transformada.

6. Aparelho, de acordo com a reivindicação5, caracterizado pelo fato de que os meios (402) para análise são configurados para esperar que cada corrente de subdados contenha uma indicação que indique a existência de informação de movimento ou não existência de informação de movimento para a respectiva camada de qualidade, e que a corrente de subdados de uma primeira dentre as camadas de qualidade contenha a informação de movimento e tenha a indicação que indique a existência de informação de movimento, ou a indicação dentro da corrente de subdados da primeira camada de qualidade indica a não existência da informação de movimento com uma parte da corrente de dados de vídeo de qualidade escalonável diferente das correntes de subdados compreendendo a informação de movimento, e esperar que a(s) corrente(s) de subdados de outra camada(s) de qualidade tenha a indicação que indique a não existência de informação de movimento.

7. Aparelho, de acordo com a reivindicação 6, caracterizado pelo fato de que os meios para análise são configurados para esperar que a corrente de subdados da primeira

camada de qualidade contenha uma indicação que indique a existência de informação de movimento, com a informação de movimento sendo igual à informação de movimento de qualidade superior ou igual a uma informação de refinamento que permita uma reconstrução da informação de movimento de qualidade superior com base na informação de movimento de qualidade inferior, e que a parte da corrente de dados de vídeo de qualidade escalonável também contenha a informação de movimento de qualidade inferior.

8. Aparelho, de acordo com a reivindicação 6 ou 7, caracterizado pelo fato de que os meios para análise são configurados de modo que a informação de movimento e a indicação se relacionem a um macrobloco da imagem.

9. Aparelho, de acordo com qualquer uma das reivindicações de 2 a 8, caracterizado pelo fato de que os meios para análise são configurados para analisar cada corrente de subdados individualmente, independentemente – com relação a um resultado de análise – da(s) outra(s) corrente(s) de subdados.

10. Aparelho, de acordo com a reivindicação 9, caracterizado pelo fato de que os meios para construção são configurados para associar a respectiva informação de coeficiente de transformada com os valores de coeficiente de transformação, com o resultado da associação sendo independente da(s) outra(s) corrente(s) de subdados.

11. Aparelho, de acordo com a reivindicação 10, caracterizado pelo fato de que uma ordem de camada é definida entre as camadas de qualidade, e a corrente de subdados de uma primeira camada de qualidade na ordem de camada permite uma

associação da respectiva informação de coeficiente de transformada com os valores de coeficiente de transformação independente da(s) corrente(s) de subdados da(s) camada(s) de qualidade subseqüente, enquanto a(s) corrente(s) de subdados das camadas de qualidade subseqüentes na ordem de camada permite uma associação da respectiva informação de coeficiente de transformada com os valores de coeficiente de transformada meramente em combinação com a(s) corrente(s) de subdados de (a) camada(s) de qualidade precedendo a respectiva camada de qualidade, onde os meios para construção são configurados para associar a informação de coeficiente de transformada de uma respectiva camada de qualidade com os valores de coeficiente de transformação pelo uso das correntes de subdados da respectiva camada de qualidade e camada(s) de qualidade precedendo a respectiva camada de qualidade.

12. Método para gerar uma corrente de dados de vídeo de qualidade escalonável (36), em que uma sequência unidimensional de valores de coeficiente de transformação é organizada em uma pluralidade de camadas de qualidade, caracterizado pelo fato de que compreende:

codificação de um sinal de vídeo (18) usando transformação em relação a bloco para obter blocos de transformada (146, 148) dos valores de coeficiente de transformação arranjados bidimensionalmente para uma imagem (140) do sinal de vídeo, onde uma ordem de exploração predeterminada (154, 156, 164, 166) com possíveis posições de exploração ordena os valores de coeficiente de transformação na dita sequência unidimensional de valores de coeficiente de

transformada; e

formação, para cada pluralidade de camadas de qualidade, de uma corrente de subdados de vídeo (30; 28, 30) contendo informação de faixa de exploração que indica um subconjunto das possíveis posições de exploração, de modo que o subconjunto de cada uma dentre a pluralidade de camadas de qualidade compreenda pelo menos uma possível posição de exploração não incluída no subconjunto de qualquer outra dentre a pluralidade de camadas de qualidade e uma das possíveis posições de exploração seja incluída em mais que um dos subconjuntos das camadas de qualidade, e informação de coeficiente de transformada nos valores de coeficiente de transformação que pertence ao subconjunto de possíveis posições de exploração da respectiva camada de qualidade, contendo um valor de contribuição por posição de exploração possível do subconjunto de possíveis posições de exploração da respectiva camada de qualidade, de modo que o valor de coeficiente de transformada de uma possível posição de exploração seja derivável com base em uma soma dos valores de contribuição para uma possível posição de exploração de mais que um dos subconjuntos das camadas de qualidade.

13. Método para reconstrução de um sinal de vídeo de uma corrente de dados de vídeo de qualidade escalonável (36) em que uma sequência unidimensional de valores de coeficiente de transformação é organizada em uma pluralidade de camadas de qualidade, a qual compreende cada pluralidade de camadas de qualidade, uma corrente de subdados de vídeo (30; 28, 30), caracterizado pelo fato de que compreende:

análise das correntes de subdados de vídeo da pluralidade de camadas de qualidade, para obter, para cada camada de qualidade, uma informação de faixa de exploração e informação de coeficiente de transformada em valores de coeficiente de transformação arranjados bidimensionalmente de diferentes blocos de transformada, onde uma ordem de exploração predeterminada com possíveis posições de exploração ordena os valores de coeficiente de transformação em uma seqüência linear de valores de coeficiente de transformação, e a informação de faixa de exploração indica um subconjunto das possíveis posições de exploração;

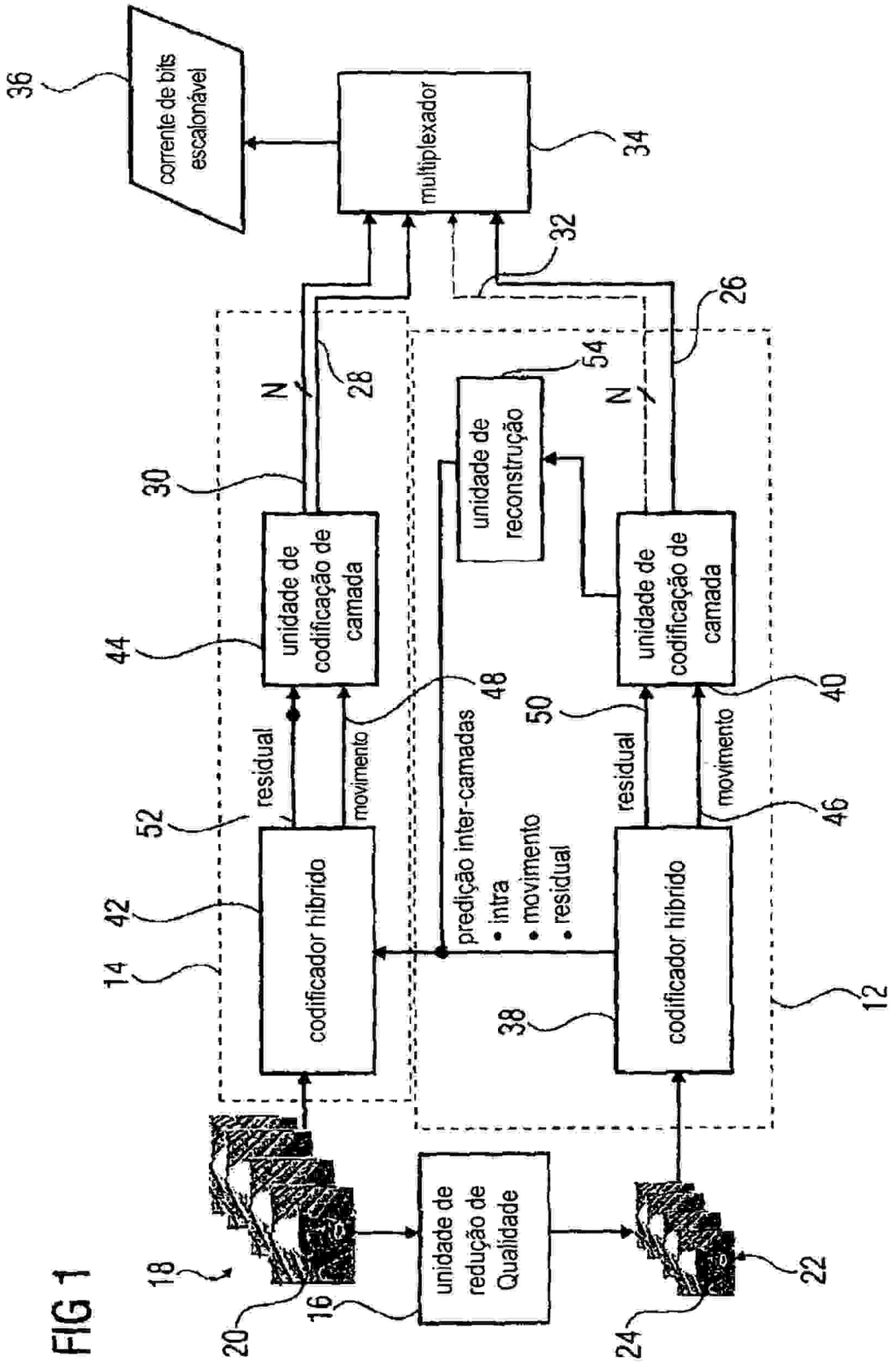
uso da informação de faixa de exploração, para cada camada de qualidade, para construir os blocos de transformada pela associação dos valores de coeficiente de transformação dos respectivos blocos de transformada da informação de coeficiente de transformada para o subconjunto das possíveis posições de exploração; e

reconstrução de uma imagem do sinal de vídeo por uma retro-transformação dos blocos de transformada, onde a análise das correntes de subdados de vídeo é executada de modo que a informação de coeficiente de transformada de mais que uma dentre as camadas de qualidade contem um valor de contribuição relacionado a um valor de coeficiente de transformação, e a construção dos blocos de transformada compreende derivar o valor para o valor de coeficiente de transformada baseado em uma soma dos valores de contribuição relacionados ao valor de coeficiente de transformada.

14. Corrente de dados de vídeo de qualidade

escalonável, em que uma sequência unidimensional de valores de coeficiente de transformação é organizada em uma pluralidade de camadas de qualidade, a qual permite uma reconstrução de um sinal de vídeo compreendendo, para cada uma dentre uma pluralidade de camadas de qualidade, uma informação de faixa de exploração e uma informação de coeficiente de transformada nos valores de coeficiente de transformada arranjados bidimensionalmente de diferentes blocos de transformada, caracterizado pelo fato de uma ordem de exploração predeterminada com possíveis posições de exploração ordena os valores de coeficiente de transformação na dita sequência unidimensional de valores de coeficiente de transformada, e a informação de faixa de exploração indica um subconjunto das possíveis posições de exploração, de modo tal que o subconjunto de cada uma dentre a pluralidade de camadas de qualidade compreende pelo menos uma possível posição de exploração não incluída no subconjunto de qualquer outra dentre a pluralidade de camadas de qualidade, onde a informação de coeficiente de transformada se refere a valores de coeficiente de transformada que pertencem ao subconjunto de possíveis posições de exploração, onde a informação de coeficiente de transformada de mais que uma das camadas de qualidade contém um valor de contribuição relacionado a um valor de coeficiente de transformada, e o valor de coeficiente de transformada de uma possível posição de exploração é derivável, com base em uma soma dos valores de contribuição para uma posição de exploração possível, de mais que um dos subconjuntos das camadas de qualidade.

FIG 1


corrente de bits escalonável
multiplexador
unidade de codificação de camada
unidade de reconstrução
unidade de codificação de camada
codificador híbrido
codificador híbrido
predição inter-camadas
• intra
• movimento
• residual
residual
movimento
residual
movimento
unidade de redução de Qualidade
N
N
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54

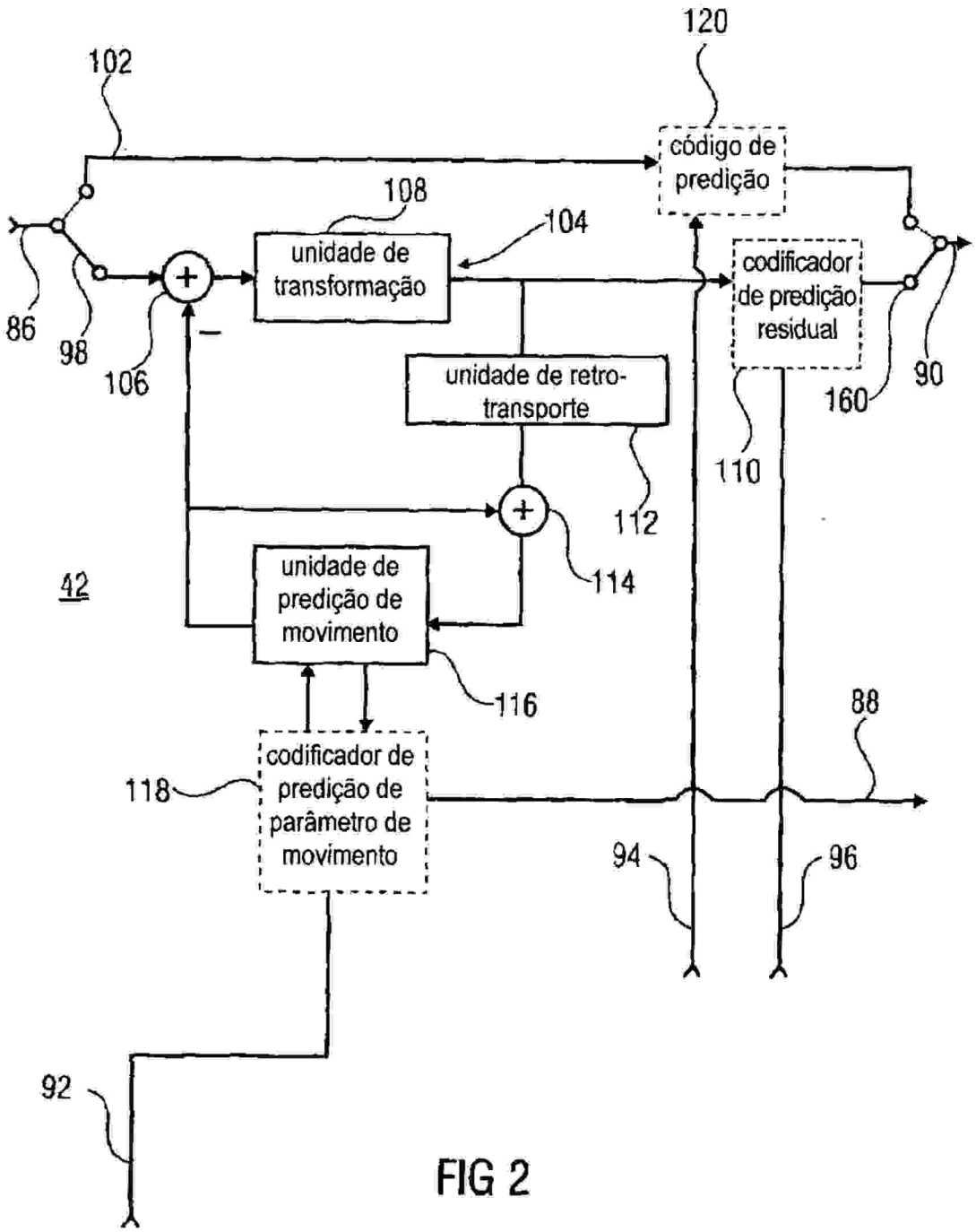

120
102
código de predição
108
104
unidade de transformação
codificador de predição residual
86
98
106
160
90
unidade de retro-transporte
110
112
114
42
unidade de predição de movimento
116
88
118
codificador de predição de parâmetro de movimento
94
96
92

FIG 2

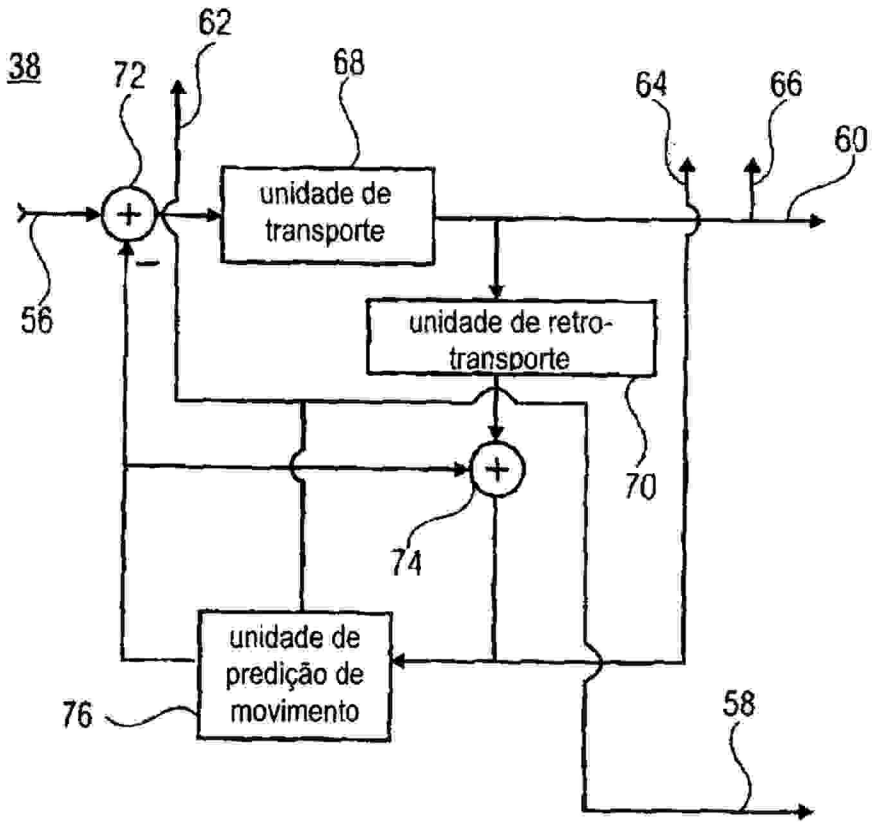

38
72
62
68
64
66
60
unidade de transporte
56
unidade de retro-transporte
70
74
unidade de predição de movimento
76
58

FIG 3

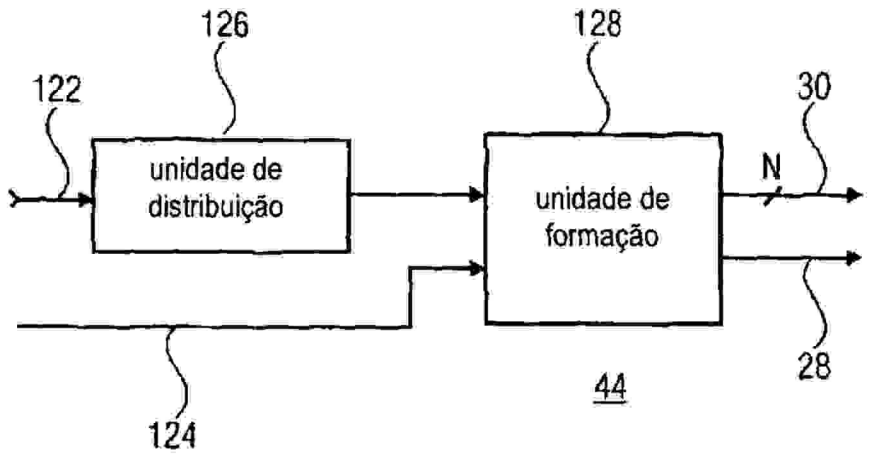

126
128
122
30
unidade de distribuição
unidade de formação
N
28
44
124

FIG 4

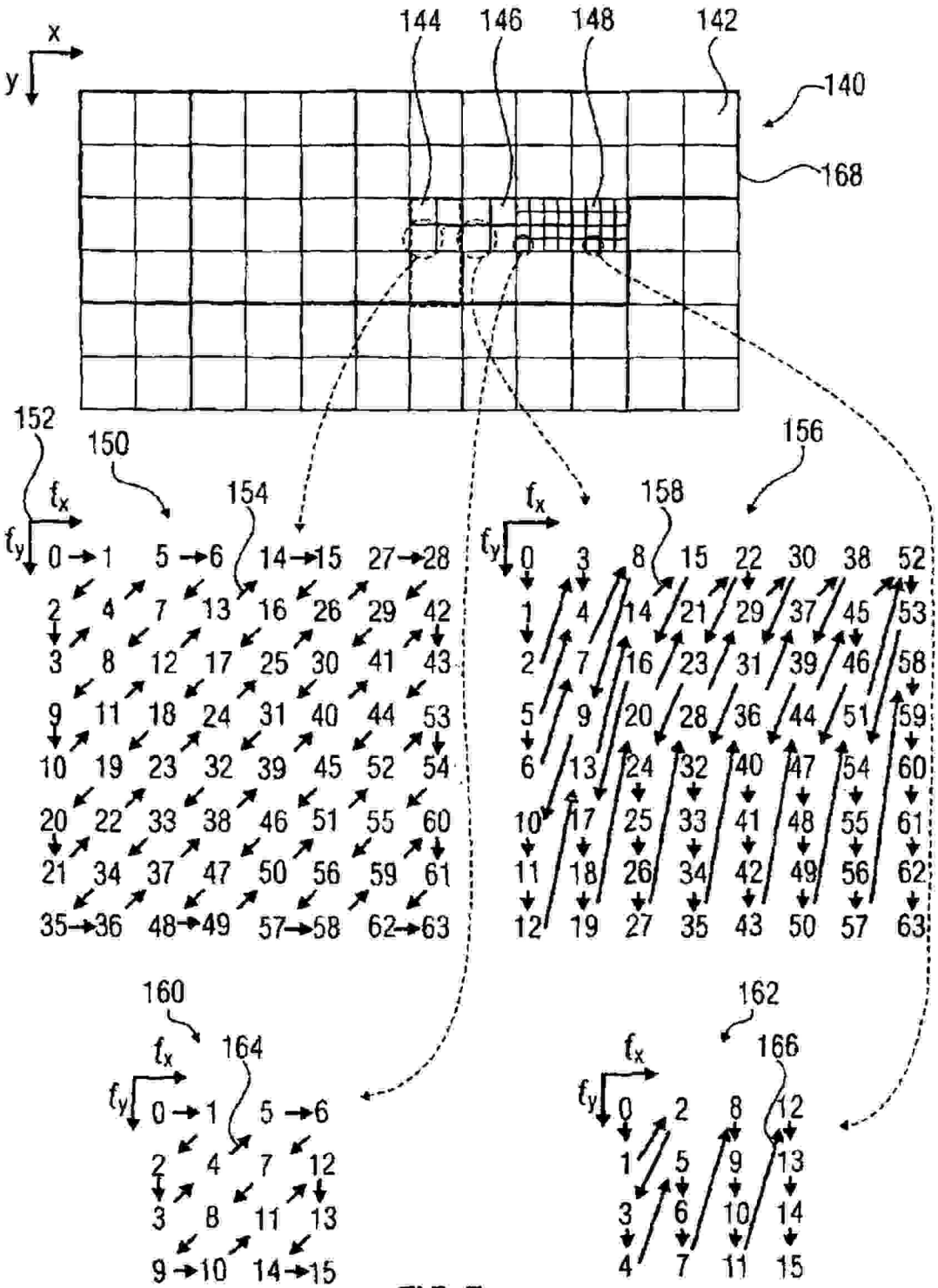
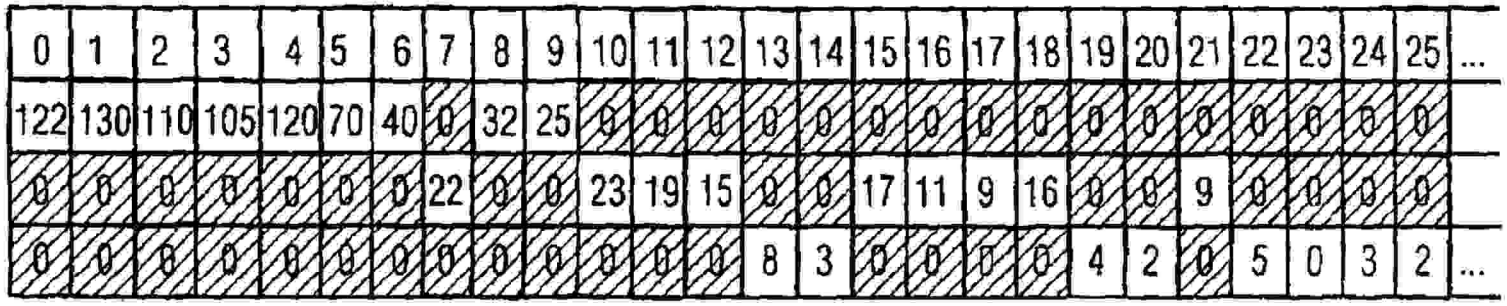

x
y
144
146
148
142
140
168
152
150
154
156
158
160
162
164
166
f_x
f_y
0 → 1 5 → 6 14 → 15 27 → 28
2 4 7 13 16 26 29 42
3 8 12 17 25 30 41 43
9 11 18 24 31 40 44 53
10 19 23 32 39 45 52 54
20 22 33 38 46 51 55 60
21 34 37 47 50 56 59 61
35 → 36 48 → 49 57 → 58 62 → 63
0 3 8 15 22 30 38 52
1 4 14 21 29 37 45 53
2 7 16 23 31 39 46 58
5 9 20 28 36 44 51 59
6 13 24 32 40 47 54 60
10 17 25 33 41 48 55 61
11 18 26 34 42 49 56 62
12 19 27 35 43 50 57 63
0 → 1 5 → 6
2 4 7 12
3 8 11 13
9 → 10 14 → 15
0 2 8 12
1 5 9 13
3 6 10 14
4 7 11 15

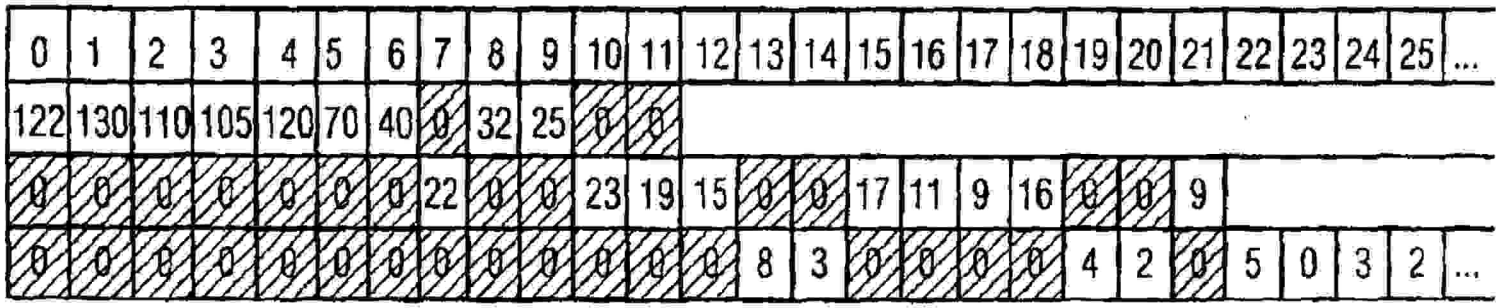
FIG 5

FIG 6A



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...
122	130	110	105	120	70	40	0	32	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	22	0	0	23	19	15	0	0	17	11	9	16	0	0	9	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	0	0	0	0	4	2	0	5	0	3	2	...

FIG 6E



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...
122	130	110	105	120	70	40	0	32	25	0	0															
0	0	0	0	0	0	0	22	0	0	23	19	15	0	0	17	11	9	16	0	0	9					
0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	0	0	0	0	4	2	0	5	0	3	2	...

FIG 6D

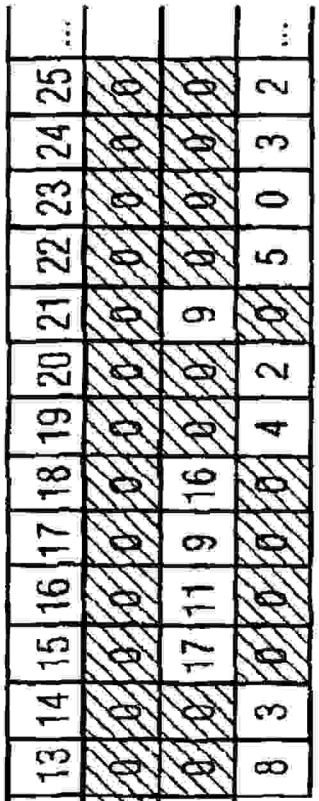

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 0 32 25 0 23 19 15 0 17 11 9 16 4 2 0 9 0 0 0 0 0
0 22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 3 2 ...

FIG 6C

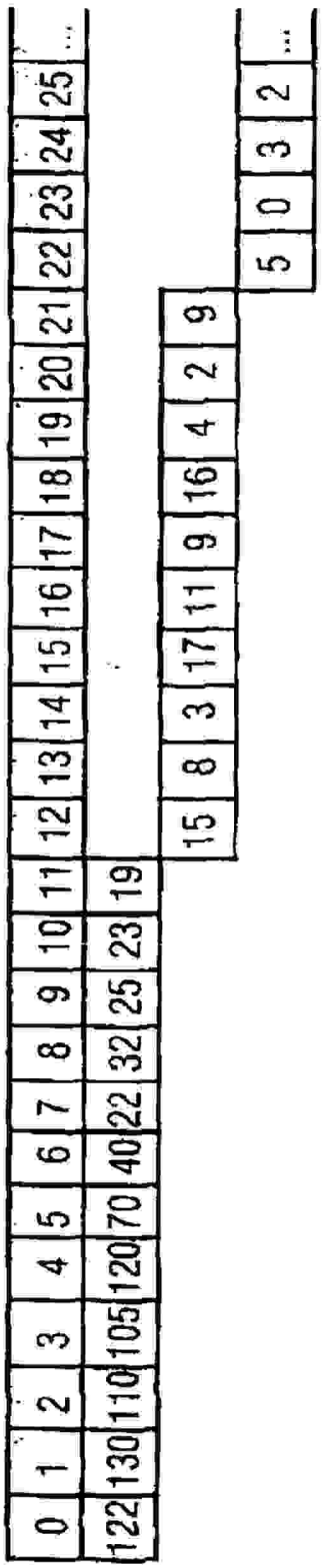

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 22 32 25 23 19
15 8 3 17 11 9 16 4 2 9
5 0 3 2 ...

FIG 6B

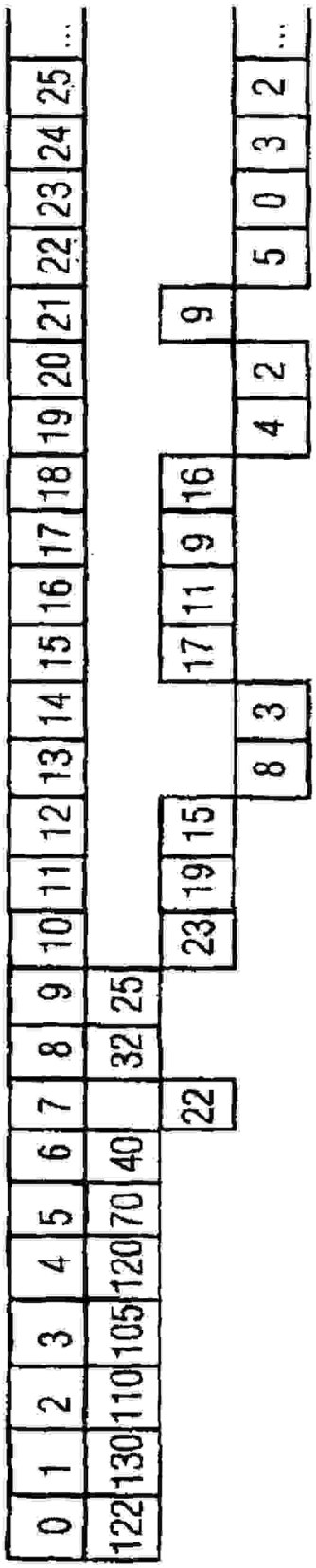

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 22 32 25 23 19 15 8 3 17 11 9 16 4 2 9 5 0 3 2 ...

FIG 6F

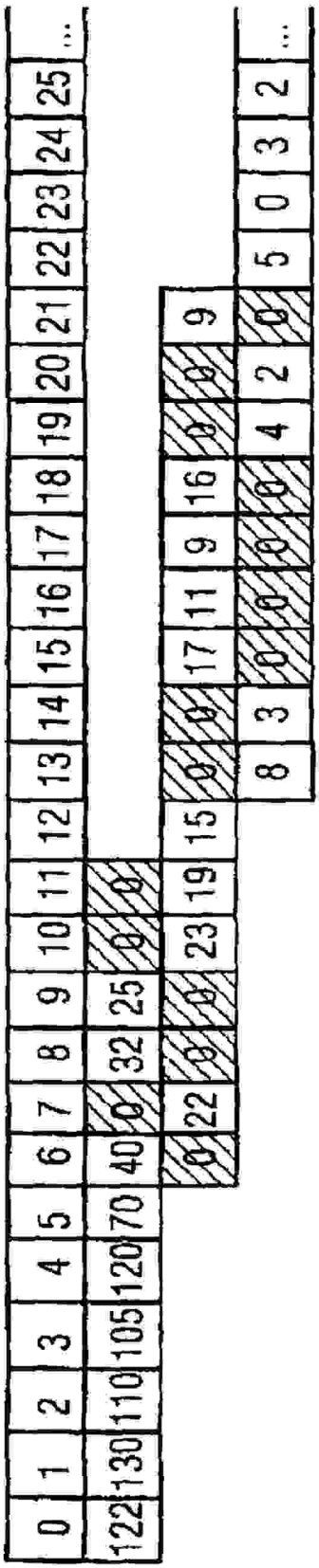
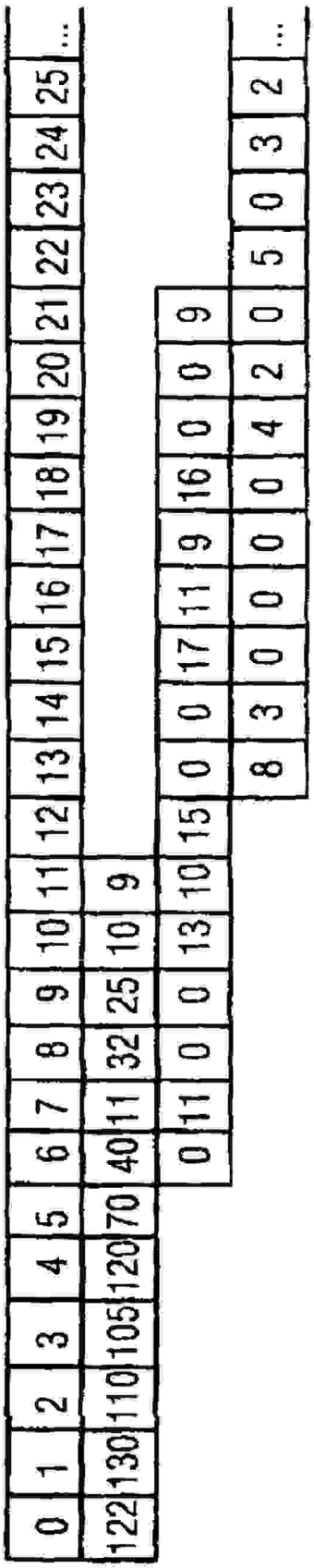

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 0 32 25 0 0 15 0 8 3 0 22 0 0 23 19 0 17 11 9 16 0 0 9 4 2 0 5 0 3 2 ...

FIG 6G


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 ...
122 130 110 105 120 70 40 11 32 25 10 9 15 0 17 11 9 16 0 0 9
0 11 0 0 13 10 0 0 0 0 0 0 0 5 0 3 2 ...
8 3 0 0 0 4 2 0

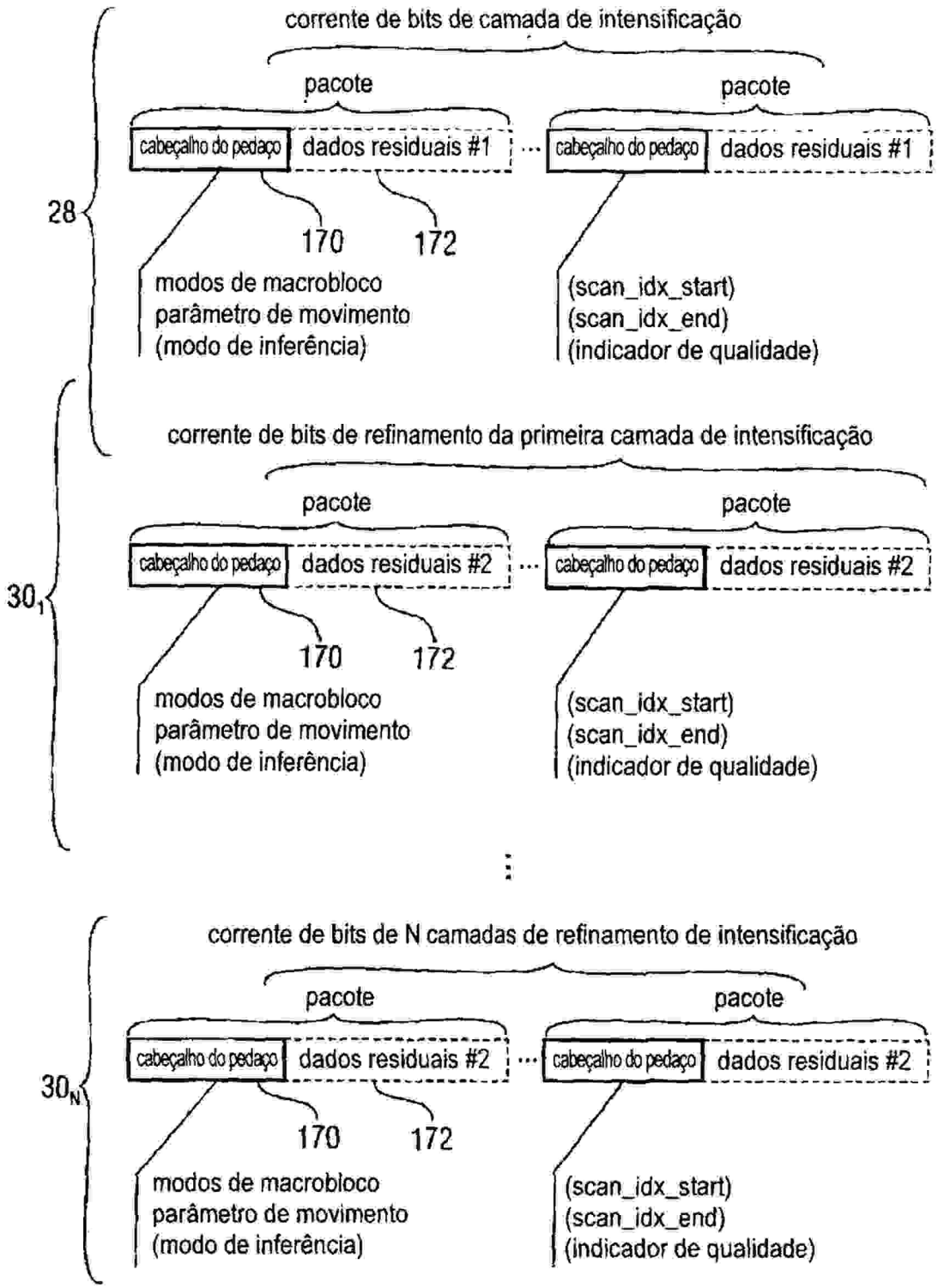

corrente de bits de camada de intensificação
pacote
pacote
cabeçalho do pedaço
dados residuais #1
cabeçalho do pedaço
dados residuais #1
28
170
172
modos de macrobloco
parâmetro de movimento
(modo de inferência)
(scan_idx_start)
(scan_idx_end)
(indicador de qualidade)
corrente de bits de refinamento da primeira camada de intensificação
pacote
pacote
cabeçalho do pedaço
dados residuais #2
cabeçalho do pedaço
dados residuais #2
30₁
170
172
modos de macrobloco
parâmetro de movimento
(modo de inferência)
(scan_idx_start)
(scan_idx_end)
(indicador de qualidade)
corrente de bits de N camadas de refinamento de intensificação
pacote
pacote
cabeçalho do pedaço
dados residuais #2
cabeçalho do pedaço
dados residuais #2
30N
170
172
modos de macrobloco
parâmetro de movimento
(modo de inferência)
(scan_idx_start)
(scan_idx_end)
(indicador de qualidade)

FIG 7

```
residual_block_cavlc( coeffLevel, startIdx, endIdx, maxNumCoeff) {
  for(i=0; i<maxNumCoeff; i++)                                        }
    coeffLevel[ i ]                                                   } 280
  coeff_token                                          <-- 240
  if( total_coeff( coeff_token) >0){                   <-- 242
    for( i=0;i<total_coef(coeff_token); i++)           <-- 244
      if(i<trailing_ones(coeff_token)){                <-- 248
        trailing_ones_sign_flag                        <-- 246
        level[ i ] =1 - 2 * trailing_onessign_flag     <-- 250
      } else {
        coeff_level                                    <-- 252
        level [ i ]=coeff_level                        <-- 254
      }
    if( total_coeff( coeff_token )<endIdx-startIdx+1){ <--256
      total_zeros                                      <--258
      zerosLeft=total_zeros                            <--260
    } else
      zerosLeft=0
    for(i=0;i<total_coeff(coeff_token)-1;i++){         <-- 262
      if( zerosLeft>0){                                <--270
        run_before                                     <--264
        run[ i ]=run_before                            <--266
      }else
        run[ i ]= 0                                    <-- 272
      zerroLeft=zerosLeft-run[ i ]                     <-- 268
    }
    run[total_coeff(coeff_token)-1]=zerosLeft          <-- 274
    coeffNum=-1                                        <--278
    for(i=total_coeff(coeff_token)-1;i>=0;i--) {                      }
      coeffNum+=run[i] + 1                                            } 276
      coeffLevel[startIdx+coeffNum]=level[i]                          }
    }
  }
}
```

FIG 8

```
residual_block_cabac( coeffLevel, startIdx, endIdx, maxNumCoeff) {
  if( maxNumCoeff==64)
    coded_block_flag=1
  else
    coded_block_flag
  for(i=0; i<maxNumCoeff;i++)                                  }
    coeffLevel[ i ]=0                                          } 322
  if(coded_block_flag){
    numCoeff=maxNumCoeffendIdx+1
    i=0starrtIdx
    do {
      significant_coeff_flag[ i ]                    <--- 312
      if(significant_coeff_flag[ i ]){               <--- 314
        last_significant_coeff_flag[ i ]             <--- 316
        if (last_significant_coeff_flag[ i ]){       <--- 318
          numCoeff=i+1                               <--- 320
        }
      }
      i++                                            <--- 310
    } while( i < numCoeff-1)
    coeff_abs_level_minus1[numCoeff-1]                         }
    coeff_sign_flag[numCoeff-1]                                } 324
    coeffLevel[ numCoeff-1 ]=                                  }
      (coeff_abs_level_minus1[numCoeff-1]+1) *                 } 326
      (1-2*coeff_sign_flag[numCoeff-1])                        }
    for( i=numCoeff-2;i>=0;i--)                    <--- 332
      if(significant_coeff_flag[ i ]) {            <--- 330
        coeff_abs_level_minus1[ i ]                            }
        coeff_sign_flag[ i ]                                   }
        coeffLevel[ i ]=(coeff_abs_level_minus1[ i ]+1)*       } 328
                        (1-2*coeff_sign_flag[ i ])             }
      }
    }
  }
```

FIG 9

FIG 10

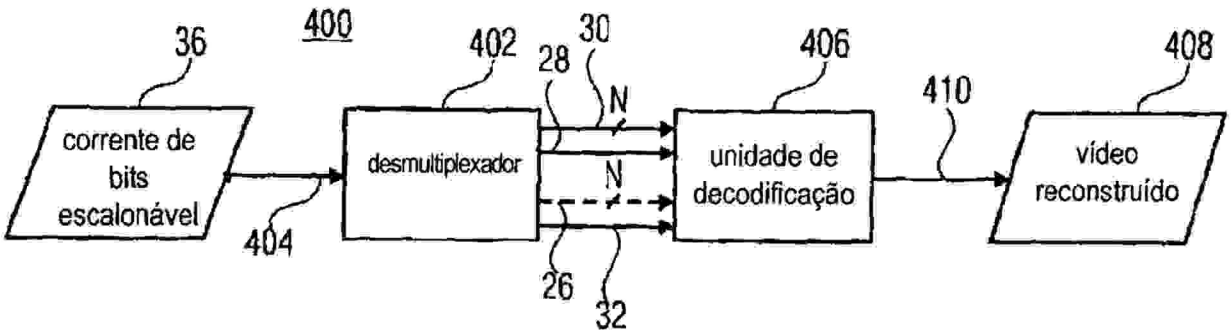
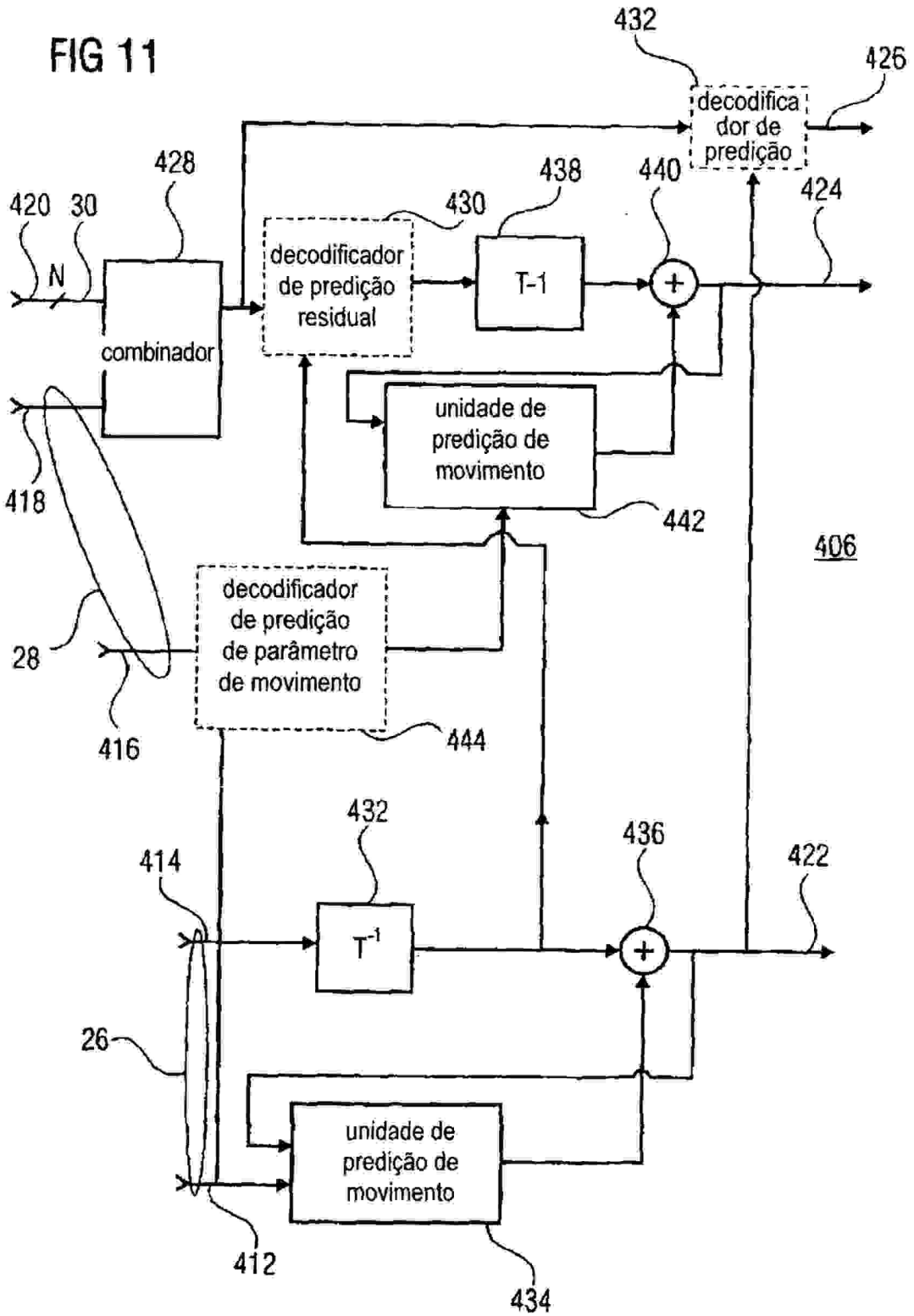

400
36
corrente de bits escalonável
404
402
desmultiplexador
28
30
N
N
26
32
406
unidade de decodificação
410
408
vídeo reconstruído

FIG 11


432
426
decodifica
dor de
predição
428
438
440
420
30
430
424
N
decodificador
de predição
residual
T-1
combinador
unidade de
predição de
movimento
418
442
406
decodificador
de predição
de parâmetro
de movimento
28
416
444
432
436
422
414
T-1
26
unidade de
predição de
movimento
412
434